NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

BATTLE GROUP STATIONING ALGEBRAIC MODELING SYSTEM: AN ANTI-AIR WARFARE TACTICAL DECISION AID METHODOLOGY

by

Donald D. Gabrielson

March 1995

Thesis Advisor: David Schrady

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE March 1995	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE Battle Group Stationing Algebraic Modeling System: An Anti-Air Warfare Tactical Decision Aid Methodology	5. FUNDING NUMBERS
6. AUTHOR(S) Gabrielson, Donald D.	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

This thesis presents a methodology that could aid the development of effective AAW (Anti-Air Warfare) ship and aircraft screens for AAW commanders at sea. The method employs stochastic discrete-event simulation of detect-to-engage sequences to develop expected leaker values for single platforms versus a common set of threats. A network flow-based stationing algorithm minimizes expected leakers by selecting the best station for each tested platform. The stationing algorithm creates AAW screen recommendations which can be evaluated using a second simulation, which gives expected leakers for a group of AAW platforms against a set of threats. The battle group simulation allows expected leaker comparisons of recommended screens with user-designed formations. Both simulations offer symbol-based graphics options. Direct application of the methodology would lead to a Tactical Decision Aid which could provide assistance in improving a naval battle group's ability to defend itself from air attack. Extensions might prove to be useful in Tactical Ballistic Missile Defense, Joint littoral AAW operations, or in land-based AAW problems.

14. SUBJECT TERMS Anti-Air Warfare, Battle Group Operations, Littoral Warfare, Tactical Decision Aid, Simulation, Optimization, Visual Basic, Command and Control, Naval Warfare, JMCIS	15. NUMBER OF PAGES * 96
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18 298-102

Approved for public release; distribution is unlimited.

BATTLE GROUP STATIONING ALGEBRAIC MODELING SYSTEM: AN ANTI-AIR WARFARE TACTICAL DECISION AID METHODOLOGY

Donald D. Gabrielson
Lieutenant, United States Navy
B.S., United States Naval Academy, 1989

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
March 1995

Author: ______________________
Donald D. Gabrielson

Approved by: ______________________
David Schrady, Thesis Advisor

____
Michael Bailey, Second Reader

____
Peter Purdue, Chairman
Department of Operations Research

ABSTRACT

This thesis presents a methodology that could aid the development of effective AAW (Anti-Air Warfare) ship and aircraft screens for AAW commanders at sea. The method employs stochastic discrete-event simulation of detect-to-engage sequences to develop expected leaker values for single platforms versus a common set of threats. A network flow-based stationing algorithm minimizes expected leakers by selecting the best station for each tested platform. The stationing algorithm creates AAW screen recommendations which can be evaluated using a second simulation, which gives expected leakers for a group of AAW platforms against a set of threats. The battle group simulation allows expected leaker comparisons of recommended screens with user-designed formations. Both simulations offer symbol-based graphics options. Direct application of the methodology would lead to a Tactical Decision Aid which could provide assistance in improving a naval battle group's ability to defend itself from air attack. Extensions might prove to be useful in Tactical Ballistic Missile Defense, Joint littoral AAW operations, or in land-based AAW problems.

TABLE OF CONTENTS

LIST OF ACRONYMS AND ABBREVIATIONS

AAW	Anti-Air Warfare
ASUW	Anti-Surface Warfare
ASW	Anti-Submarine Warfare
ATP	Allied Tactical Publication
CAP	Combat Air Patrol
CIWS	Close-In Weapons System
CV	Aircraft Carrier
CVBG	Aircraft Carrier Battle Group
deg	degrees (angular)
DTE	Detect -to-Engage
IREPS	Integrated Refractive Effects Prediction System
JMCIS	Joint Maritime Command Information System
kts	Knots, nautical miles per hour
Kft	Thousands of feet
M	Mach number (as a fraction of the speed of sound)
MOE	Measure of Effectiveness
MS-DOS	Microsoft Disk Operating System
NDP	Naval Doctrine Publication
NM	Nautical Mile
NTDS	Naval Tactical Data System
NWP	Naval Warfare Publication
RCS	RADAR Cross Section
SAM	Surface to Air Missile
TTI	Time to Intercept
YDS	Yards
ZZ	Zulu-Zulu, the center of a formation

EXECUTIVE SUMMARY

Naval Battle Group commanders face the challenge of operating in a littoral environment rich with threats. Anti-Air Warfare (AAW) plays a major role in the littorals; aircraft and Anti-Ship Cruise Missiles (ASCMs) have become a viable substitute for ships in protecting a coastline. Developing nations eager to assert their influence have found a relatively cheap way to build a significant capability to defend their shores, with a flexible force that also can contribute to overland combat forces.

This reality, when coupled with diminishing force structures and increased presence roles, has changed the nature of naval warfare. The AAW force we have constructed depends heavily on an ability to prevent enemy weapons from striking their mark, shaping our operational doctrine into one which does not lend itself easily to the littoral environment. Where we used to count on early warning and battlespace dominance, we now must be prepared for unanticipated attacks. Concealment, deception, and battlespace dominance have been considerably reduced, limiting operational choices to a requirement to maximize the defensive capabilities of a high-value battle group through maneuver and weapons system use. Effective and efficient weapons system operation has become more critical than ever.

At issue is how to station AAW platforms in order to maximize the air defensive capabilities of a battle group. Blue-water tactics do not work well in the littorals; reducing the number of ships available in each battle group further complicates defense. Unfortunately, no tools of significant value exist that aid commanders at sea in assessing their AAW strengths and weaknesses. Formations must be selected using rules-of-thumb and personal preference, with little or no numerical analysis available to support tactical decisions.

This thesis presents a new method for developing AAW formations. It uses computer simulation to model and estimate AAW system performance of individual platforms versus a set of threats. Combatant ships and tactical fighter aircraft are both included in our model. Our model uses readily-available system characteristics, avoiding the use of performance estimates. Because of this, we can test the effects of platform losses or capability changes, or threat changes. We have developed a method which allows decision makers to evaluate platforms,

battle groups, and stationing options in a numerical fashion, and will produce data about the AAW effectiveness levels offered by any conceivable formation.

Platforms are tested in user-specified locations, and their performance data is used by a stationing algorithm to generate an effective AAW formation. The stationing algorithm formation can be evaluated and compared to any other formation using a battle group simulation model. Using this methodology, decision makers can be presented with useful data related to operational choices. Decisions can be supported by concrete, numerical analysis.

Our methodology is developed and presented in the first half of this thesis. The second half presents a prototype tool we developed. It operates on any desktop PC equipped with the Microsoft *Windows 3.1* operating system, and is a fully-functioning demonstration tool. It allows users to build a threat set and test it against a set of AAW platforms which also are user-defined. Graphics are available to monitor the simulations; our simulations follow a Detect-to-Engage (DTE) sequence which will be familiar to anyone conversant in AAW concepts.

We also present an example scenario. It uses unclassified data, and was developed to illustrate the flexibility of our method. We test each platform individually in our battle group, then use the stationing algorithm to develop an effective formation. Following this, we test our formation using the second simulation, which accounts for interactions present in battle group AAW. We conduct sensitivity analysis by removing AAW platforms and investigate the effects of restationing ships after a loss. We also test the battle group for saturation sensitivity by incrementally increasing the threat density.

Our results are encouraging. We can solve the stationing problem in a reasonable amount of time. We believe that it would make an excellent tactical decision aid (TDA), offering battle group and AAW commanders a useful tool which could provide numerical insight in a real-world environment. The building blocks for a TDA using our methodology already exist in the JMCIS (Joint Maritime Command Information System) operating environment currently deployed throughout the U.S. Navy. JMCIS is a system which continues to grow; our methodology could contribute to a new area of growth in at-sea simulation and modeling capabilities.

I. INTRODUCTION

> The complexity of modern warfare in both methods and means demands exacting analysis of the measures and countermeasures introduced at every stage by ourselves and the enemy. . . Each type of naval operation had to be analyzed theoretically to determine the maximum potentialities of the equipment involved, the probable reactions of the personnel, and the nature of the tactics which would combine equipment and personnel in an optimum manner.
>
> --Admiral E.J. King, 1945

A. TACTICAL BACKGROUND

Current U.S. naval strategy follows guidelines drawn by the Chief of Naval Operations and the Joint Chiefs of Staff, detailed by the documents respectively entitled *Naval Warfare* [Ref. 1] and *Joint Warfare of the U.S. Armed Forces* [Ref. 2]. Both highlight a requirement for naval forces to operate in near-land areas referred to as *littoral regions.* This shift away from a deep-water strategy primarily reflects the transition toward countering a growing number of regional military threats, few of which have an ability to operate capably on the open ocean. Naval forces have shifted emphasis toward strike missions; with that shift comes a requirement to operate as close as possible to land, allowing deeper penetration into enemy territory and a larger selection of targets.

Central to contemporary naval operations is the aircraft carrier battle group (CVBG), which typically consists of an aircraft carrier (CV) and six to nine armed surface (*combatant*) ships [Ref. 3]. The primary mission of CVBGs operating in littoral regions has become projection of power ashore in support of or in place of land-based forces [Ref. 1]. Although other significant threats exist in littoral regions, this thesis focuses on countering the airborne threats present in littoral areas. Since an aircraft carrier's mission of projecting power ashore consumes a large portion of its combat-capable resources, the remaining combatants and fighter aircraft (called combat

air patrols (*CAP*)), assume the primary role of defending the CV from air attack. This mission, defensive in nature, is referred to as anti-air warfare (*AAW*).

In order to assume an effective AAW posture, escorting ships and CAP are placed in locations centered around the CV in an effort to allow the best opportunity to shoot down attacking enemies. A specific pattern is referred to as a *screen, formation,* or *disposition* interchangeably. The formation selected depends upon the potential enemy weapons faced (called *order of battle*), geographical and scheduling concerns, the mix of surface combatants and aircraft present, and other tactical considerations expressed by the naval force commander. Escorts might operate in a single location or be assigned to operate in an area, or *sector*; both are located relative to the CV.

Formation choice typically depends on the mix of combatants and CAP available, the operating area, and the capabilities of enemy forces. A force commander has two doctrinal options when selecting a formation to use: AAW platforms could be located as far as possible along a potential threat's flight path (called a *threat axis*), or they could be stationed close to the protected unit. Each approach has merits and drawbacks: the first allows more time to react before launch platforms can reach adequate firing positions, but it gambles on the likelihood that an attack will approach from the direction(s) covered. The second option counters an attack in progress, but also allows greater opportunity for an enemy to locate and attack the protected unit. Simply stated, the first screen philosophy aims to prevent successful attacks, and the second seeks to minimize their effectiveness. An AAW axiom refers to shooting the archers, or, missing them, catching their arrows. Once an attacker has fired, the defender's emphasis shifts to preventing enemy weapons from finding their mark. In practice, a force commander usually chooses a formation which represents a mix of each tactical philosophy. This allows an opportunity to dilute an attacking force while still offering protection from threats which survive the outer defense platform weapons. This concept is referred to as *Defense-in-Depth* [Ref. 4].

During littoral operations, a commander faces greater demands on his ability to react quickly and correctly to attacks. Because his forces must operate close to land, opportunity for attack from any direction is significantly increased. In contrast, during operations far from land (*blue-water operations*), attackers face the challenges of locating, targeting, and striking targets far from friendly fuel sources. This typically limits the possibilities for attacks to roughly straight-in approaches from operating bases. Open water allows nearly any sensible formation, and ships conducting blue-water operations have a significant ability to remain undetected through maneuver, deception, and detectable emissions control discipline. Normally, a blue-water force can maximize its defenses along the threat axes, providing a substantial level of security through stealth, early warning, and an ability to foil a large portion of an attack before it can reach a firing position.

Littoral warfare provides little security of these forms. Reduced operating area size constrains a commander's formation options, forcing much more closely-spaced screens. Ships are easily tracked, since the omnipresent threat of attack requires near-constant operation of detectable warning sensors. Land-based anti-ship missiles become a viable attack option for enemy forces. Since fuel constraints are much less a factor due to the short ranges involved, airborne attackers enjoy a much wider range of choices for avenues of approach. Furthermore, the short ranges coupled with the high speeds of modern combat significantly reduce a naval force's ability to predict and counter an attack-- decisions are allowed seconds as opposed to tens of minutes or more in blue-water operations.

The proliferation of high-technology weapons systems has radically altered our ability to operate in littoral waters without carefully considering the threats. Small, fast, low-flying missiles have significantly increased the probability of success in attacking ships at sea. Commonly, attackers can launch cruise missiles from low altitudes, at distances which significantly reduce the chances of losing valuable launch platforms. A naval force operating in littoral waters faces a threat which might approach from any

direction, day or night, launch its weapons, and flee with little chance of being effectively countered. This places a high premium on the defensive capabilities of our naval forces. Reducing the number of enemies that overcome friendly defenses (called *leakers*) is the goal of anti-air warfare. Contemporary expectations place an extremely high demand on our military's ability to nearly guarantee no losses before committing to combat. Protection from leakers has become a requirement, yet no tools are available at sea to estimate their likelihood. While tools that analyze individual platform sensor coverage exist, there does not exist a useful planning aid which provides substantial insight into how to best station AAW platforms when countering specific threats. Planners need an ability to compare formation options in order to decide how to employ available AAW assets.

B. PROBLEM DESCRIPTION AND APPROACH

The problem at hand is to determine the best locations for AAW platforms, given their engagement capabilities, the predicted threat axes, and the predicted threat density. This requires a method which recognizes and capitalizes on the strengths of available platforms, and a measure of effectiveness (MOE) which allows comparisons of value. We chose to minimize the expected number of leakers by assigning AAW platforms to their most effective stations for countering the predicted threat.

We decided to model the AAW detect-to-engage (DTE) sequence for each available platform in a predetermined set of stations, using a predetermined set of threats. We will simulate the operation of each platform in each feasible location. Next, we solve an assignment problem based on the measured performance of each platform in each station. Following this, we will evaluate the solution using a simulation which captures synergistic effects not modeled in the single-platform DTE simulation.

C. THESIS OUTLINE

The remainder of this thesis provides technical requirements and describes a prototype tool which uses our methodology. Chapter II presents simulation requirements and the assignment model formulation. Chapter III introduces a prototype tool created using our methodology. Chapter IV demonstrates the use of the prototype. The last chapter provides conclusions and recommendations for development of a deployable Tactical Decision Aid (TDA).

II. METHODOLOGY DESCRIPTION

A. REQUIREMENTS

The model must be able to convert data equating AAW platform capabilities and threat parameters into a ranked list of desired configurations for the platforms. It must be flexible, with provisions to predict the effects of changes in capabilities in both organic platforms and predicted threats.

B. ASSUMPTIONS

Using individual platform AAW effectiveness data, ignoring battle group interaction effects, the stationing problem can be solved using optimization techniques. A formation recommendation can be developed by maximizing each platform's AAW performance, obeying a set of constraints. Unfortunately, solving the problem as a sequence of individual leaker minimizations will not allow a mathematically optimal placement. But it does allow a fast solution. We selected this approach, hoping that the locally optimal solutions would be sufficiently close to global optimality for real-world use.

Numerical AAW performance data for individual platforms in various stations can be created using discrete-event stochastic simulation. A DTE simulation can produce statistical estimations of a platform's AAW performance in tested stations. Expected number of leaker values for each platform in each station can be used by a stationing algorithm. The stationing algorithm would select the best possible locations for each platform tested, according to a set of constraints governing feasible locations.

The assumptions required for data generation using a single-platform DTE simulation are:

1. Each station is a single point.
2. Stations tested will be allowed in the stationing model solution.
3. Threats concentrate their effort on a single target.
4. The simulation follows a time-based format.

5. Round-earth geometry will be used.

6. Random outcomes will be generated.

The assumptions required for the stationing problem are:

7. Each platform can occupy only one station.

8. No more than one ship may occupy a single station.

9. Each platform will be located in the best possible station from which it can operate independently of any other considered units.

These assumptions exist to simplify the creation of a tractable model, and to inject the basic considerations of a real-world decision-maker. Assumption (1) eliminates averaging of simulation data over geographical areas, allowing a determination of the relative values of points within sector stations. Assumption (2) ensures that the stationing process considers only those stations which will be allowed in the final screen. Assumption (3) allows worst-case attack modeling. Assumption (4) creates a model analogous to real-world DTE sequences. Assumption (5) is required due to the effects of altitude and range on the DTE sequence. Assumption (6) reflects the fact that real-world DTE sequences are not deterministic. Assumption (7) ensures that no platform is assigned more than one station. Assumption (8) ensures that no two platforms occupy the same station. Assumption (9) recognizes that in the event of a platform loss, the remaining platforms need to be in positions which require the least possible re-stationing to cover the loss. Assumptions (1) through (6) become the central features of the DTE simulations. Assumptions (7) through (9) become the core features of the stationing algorithm.

C. FUNCTIONAL STRUCTURE

Figure (1) illustrates our methodology structural template. Our method requires two simulations and a stationing algorithm. Input data are created for the simulations using two platform databases, a threat set editor, and a station set editor. One database contains AAW platforms; the other contains threat vehicles. The single-platform DTE simulation produces expected leaker data for the stationing algorithm, which produces a formation. The battle group

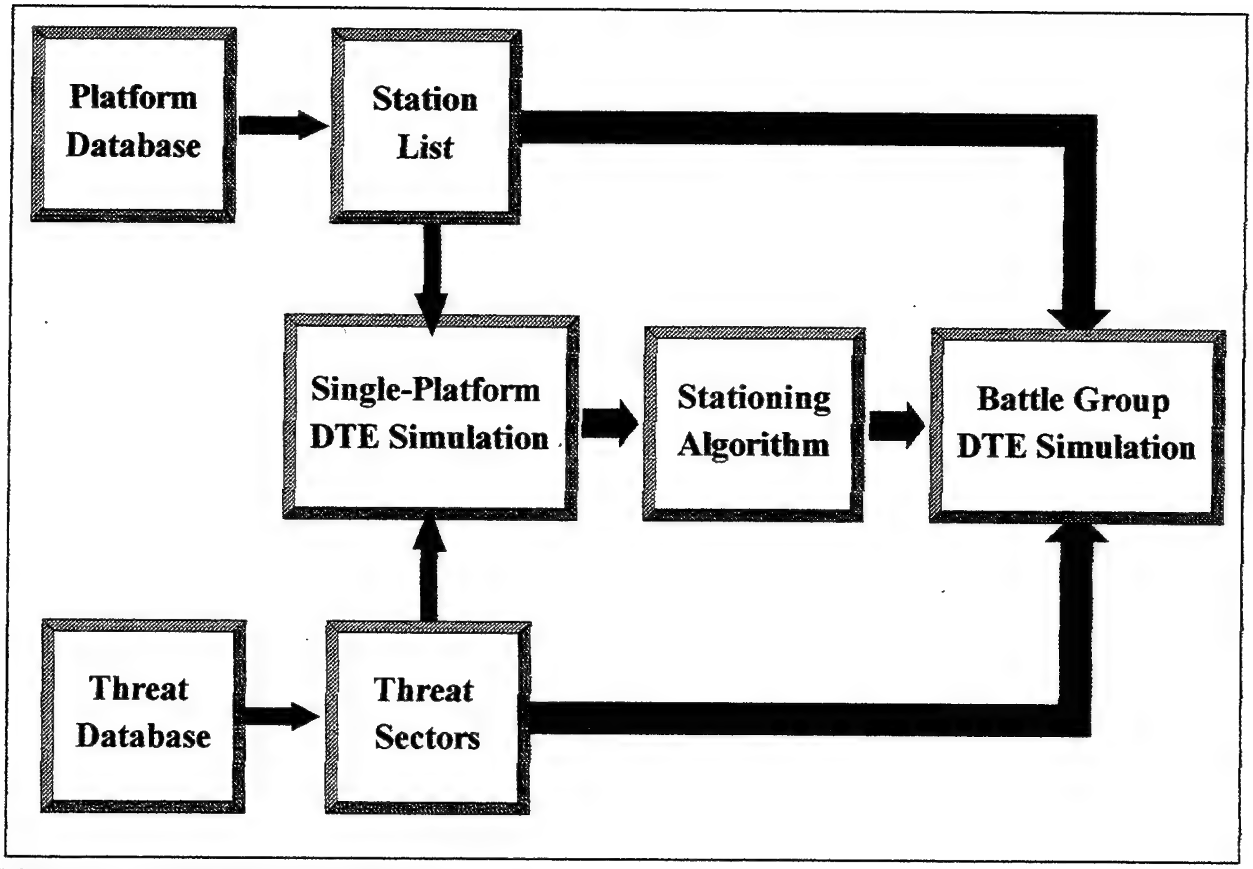

Figure (1). Methodology Block Diagram.

DTE simulation provides an estimation of battle group AAW effectiveness, allowing comparison among any number of feasible formations.

D. SINGLE-PLATFORM DTE SIMULATION

The single-platform DTE simulation produces expected leaker data which can be used by the stationing algorithm. Assumptions (1) through (6) in (B) apply. For each threat, four random variables exist: threat approach bearing, detection time, engagement time, and engagement outcome. The last three are dependent variables; their outcomes are conditioned on earlier results. This leads to a stochastic, discrete-event simulation model.

Simulation data quality will influence the stationing algorithm's results. Two factors in the simulation affect data quality. Fidelity plays a primary role. Second, the statistical quality of the data is also important. Statistical quality can be measured in terms of the variance of the

estimated expected leaker values. In m replications, Let x_s be the number of leakers for a single replication, s. Then:

$$\overline{x_m} = E[\text{Leakers}] = \tfrac{1}{m}\sum_{s=1}^{m} x_s \tag{2.1}$$

$$S_m = S(\text{Leakers}) = \sqrt{\tfrac{1}{m-1}\sum_{s=1}^{m}(x_s - \bar{x}_m)^2} \tag{2.2}$$

$$cv_m = \frac{S_m}{\overline{x_m}\sqrt{m}} \tag{2.3}$$

Equation (2.1) defines the expected value, or average number, of leakers for a platform in a station over a series of replications using the same set of threats. Equation (2.2) defines the estimate of the standard deviation of each x_s. Equation (2.3) is the estimate of cv_m of $\overline{x_m}$, the coefficient of variance of the expected number of leakers, related to m replications [Ref. 5]. We choose m such that cv_m is sufficiently small, so that $\overline{x_m}$ is an estimate whose quality we control. The simulation produces the following run statistics for each platform in each station:

1. E[Leakers], the expected number of leakers.
2. *cv*, the Coefficient of Variation of Leakers.
3. E[SAMS fired], the expected number of SAMs fired.
4. E[Time], the expected battle length.

We use the expected number of leakers in the stationing algorithm as a figure of merit for comparing platforms in various stations. The expected number of SAMs fired gives an estimation of the number of SAMs required to achieve similar results under the same conditions. The expected battle length provides a method to measure battle pace, although our model does not account for intelligence cueing or early warning.

E. FIXED PARAMETERS

1. AAW Platforms

We model the following fixed parameters for each AAW platform:

1. RADAR search interval (seconds between detection sweeps)
2. RADAR mast height (ft)
3. RADAR 95% P_d detection range (NM for 1 m^2 target)
4. SAM magazine size
5. SAM firing rate (seconds between consecutive firings)
6. MAXMIF (maximum simultaneous missiles in flight)
7. SAM flight characteristics
    a. max range (NM)
    b. min range (NM)
    c. speed (kts)
    d. max altitude (Kft)

The fixed parameters represent measurable system characteristics. RADAR detection ranges use range predictions created by the IREPS (Integrated Refractive Effects Prediction System) model currently in use throughout the U.S. Navy. Non-homogenous environment effects such as ducting or land-sea interface effects were not modeled. Because of time constraints, we included only one weapons system for each platform, which is assumed to be the longest-range weapon available to that platform. Platforms could be ships or aircraft. For brevity, *SAM* denotes a generic AAW missile, whether ship- or aircraft-fired. Figure (2) presents a diagram of fixed AAW platform parameters.

2. Threat Vehicles

Threat vehicles also might be of different types, requiring a model which incorporates the threat parameters which influence the DTE sequence. We include the following fixed parameters:

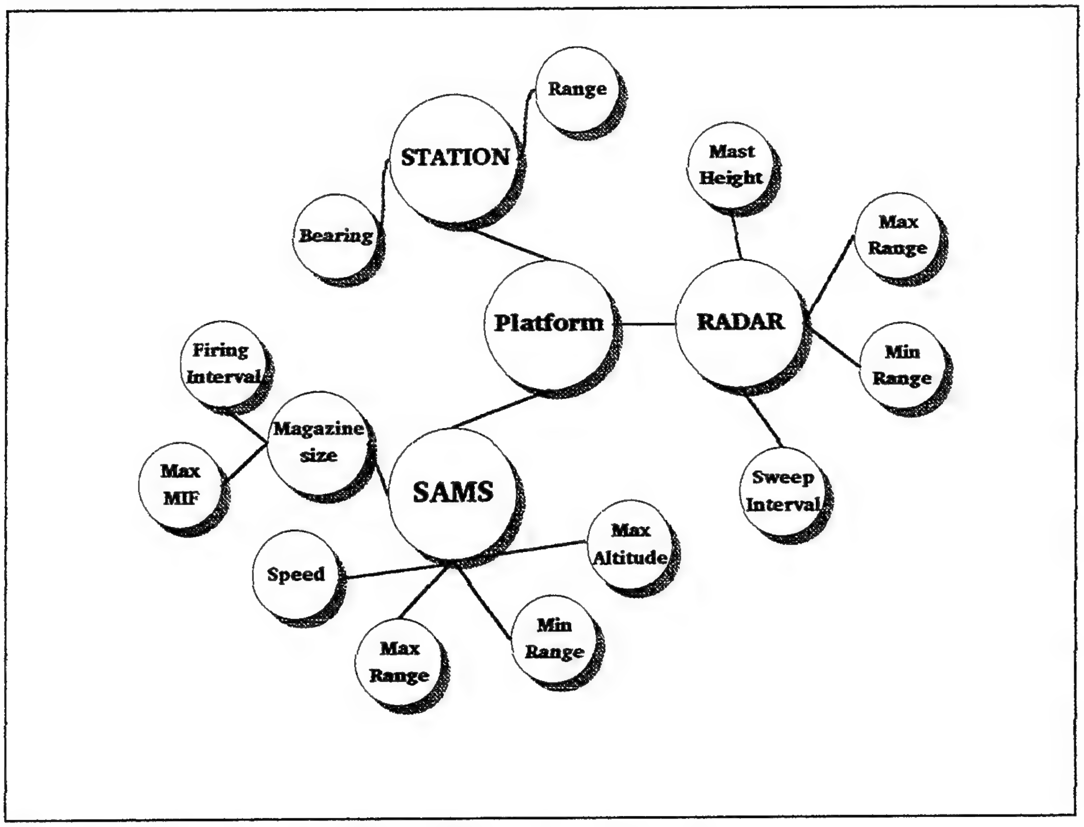

Figure (2). Simulation AAW platform parameters.

1. Threat launch time (seconds after problem start)
2. Threat speed (kts)
3. Threat RCS (m^2)
4. Threat flight profile
    a. Launch range (NM)
    b. Launch altitude (ft)
    c. Midcourse range (NM)
    d. Midcourse altitude (ft)
    e. Terminal Range (NM)
    f. Terminal altitude (ft)
    g. Final range (NM)
5. Threat sector

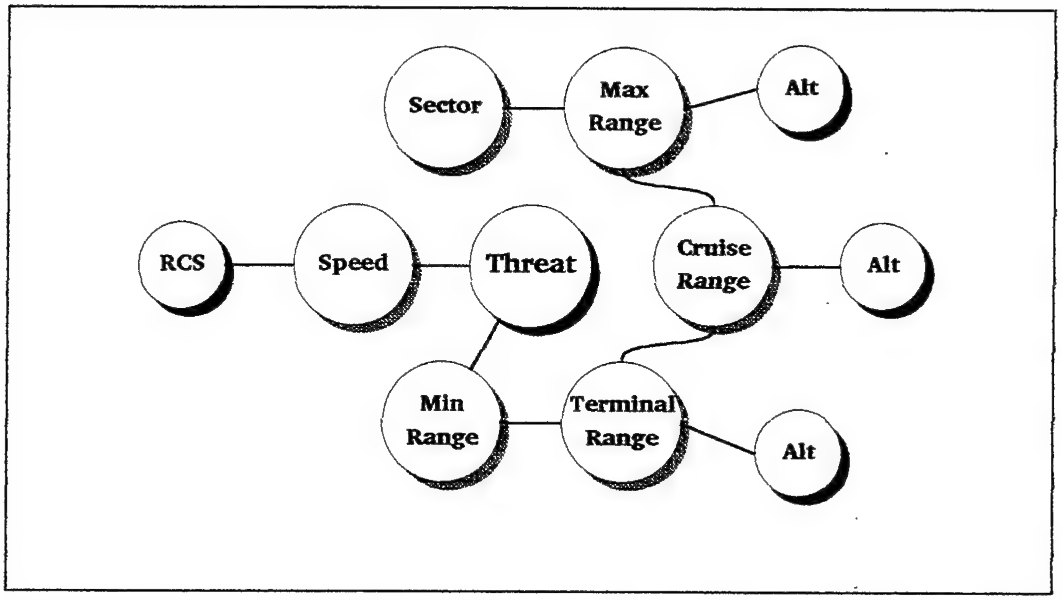

Figure (3). Simulation threat parameters.

a. Left sector bearing (deg)

b. Right sector bearing (deg)

Figure (3) presents a diagram of fixed threat vehicle parameters. Fixed, user-determined threat launch times and speeds establish a queue of threats which AAW platforms must engage. We use fixed launch intervals for each platform, based on maximum SAM range. For weapons with a range of at least 50 NM, a the firing interval is 3 seconds. Weapons with a range of less than 20 NM have a firing interval of 2 seconds. Weapons with a range between 20 and 60 NM use a firing interval of 8 seconds. The remaining parameters affect the random outcomes present in the DTE sequence. Random parameters and the resulting variables are presented next.

F. ENGAGEMENT OUTCOME PARAMETERS

We use three parameters:

1. P_d is the probability that a target will be detected by a specified platform on a single sensor sweep, given that it is above the sensor's horizon, determined by sensor height and threat altitude.
2. P_k is the probability that a SAM will destroy its target, given sufficient detection conditions for engagement and a successful intercept, which requires continued detections until the time-to-intercept (TTI).
3. User error rate is the probability that a SAM will be fired, even though it will not be detectable at TTI.

Since no closed-form mathematics exist to accurately calculate P_d and P_k, we developed parametric density functions for P_d and P_k using empirical data and qualitative system characteristics. Their results have not been accredited or compared with accredited models, but we believe that they produce predictions reasonably close to real-world system trends, at least for the purposes of our prototype model.

Error rate is fixed at a ten percent probability of firing a SAM which cannot intercept its target. Its parameter follows a uniform (0,1) distribution. P_d and P_k densities reflect the fact that detection and kill probabilities vary with geographical relationships, AAW systems, and threat characteristics. The P_d function is presented in mathematical form as:

speed = threat speed (kts)

RCS = threat RCS (m^2)

alt = threat altitude (ft)

radar = .75 * maxradar (NM, where maxradar is the platform parameter for .95 P_d)

$$P_d = P(\text{detect} \mid \textit{above horizon}) = .95 \bullet \left(1 - \sqrt{\frac{speed}{10000 \bullet RCS \bullet alt}}\right) \bullet \left(1 - \left(\frac{range}{radar}\right)^3 \bullet \left(\ln\left(\frac{range}{radar}\right)\right)^3\right) \quad (2.4)$$

Equation (2.4) is used whenever a detection sweep is made by a platform, calculated independently for each threat and compared to a uniformly distributed random number. This

gives an instantaneous detection condition for each threat on each sensor sweep. If a threat is engaged, the scaling factor of 0.95 shifts upward to 0.99, reflecting a higher amount of effort being placed on tracking engaged threats. Figures (4) through (7) give graphical examples of families of curves generated by the detection density function.

The P_k function is presented in mathematical form:

speed = threat speed (kts)

RCS = threat RCS (m^2)

alt = threat altitude (ft)

SAM = maximum SAM effective range (NM)

SysPk = effectiveness level of a SAM system

$$P_k = P(\text{kill} \mid \text{detected, within env}) = \frac{SysPk}{100} \bullet \left(1 - \sqrt{\frac{speed}{2 \bullet SAM \bullet RCS \bullet alt}}\right) \bullet \left(1 - \left(\frac{range}{SAM}\right)^3 \bullet \left(\ln\left(\frac{range}{SAM}\right)\right)^3\right) \quad (2.5)$$

Equation (2.5) is used to determine the outcome of each engagement independently, in the same manner as detection outcomes. *SysPk* is an integer-valued scaling coefficient associated with each AAW platform. It allows a measure of control over the ordinate of the major inflection point in the P_k curve. At the maximum effective SAM range, the P_k curves will have an upper bound specified by *SysPk*/100.

In order to reflect the better performance of short-range SAM systems against low-flying, small-RCS targets, we increase P_k for SAM systems with a maximum range of less than 20 NM by first finding the P_k shown in (2.5). We then calculate the fourth root of that P_k to give an upwardly-adjusted value. Figures (8) through (11) give graphical examples of families of curves generated by the kill density function. Examples use a *SysPk* value of 95. Each discrete altitude creates an individual curve. Altitudes plotted are from 10 to 1000 feet, in 10-foot increments. P_k increases with altitude; as a result, the lowest P_k curves correspond to the lowest altitudes.

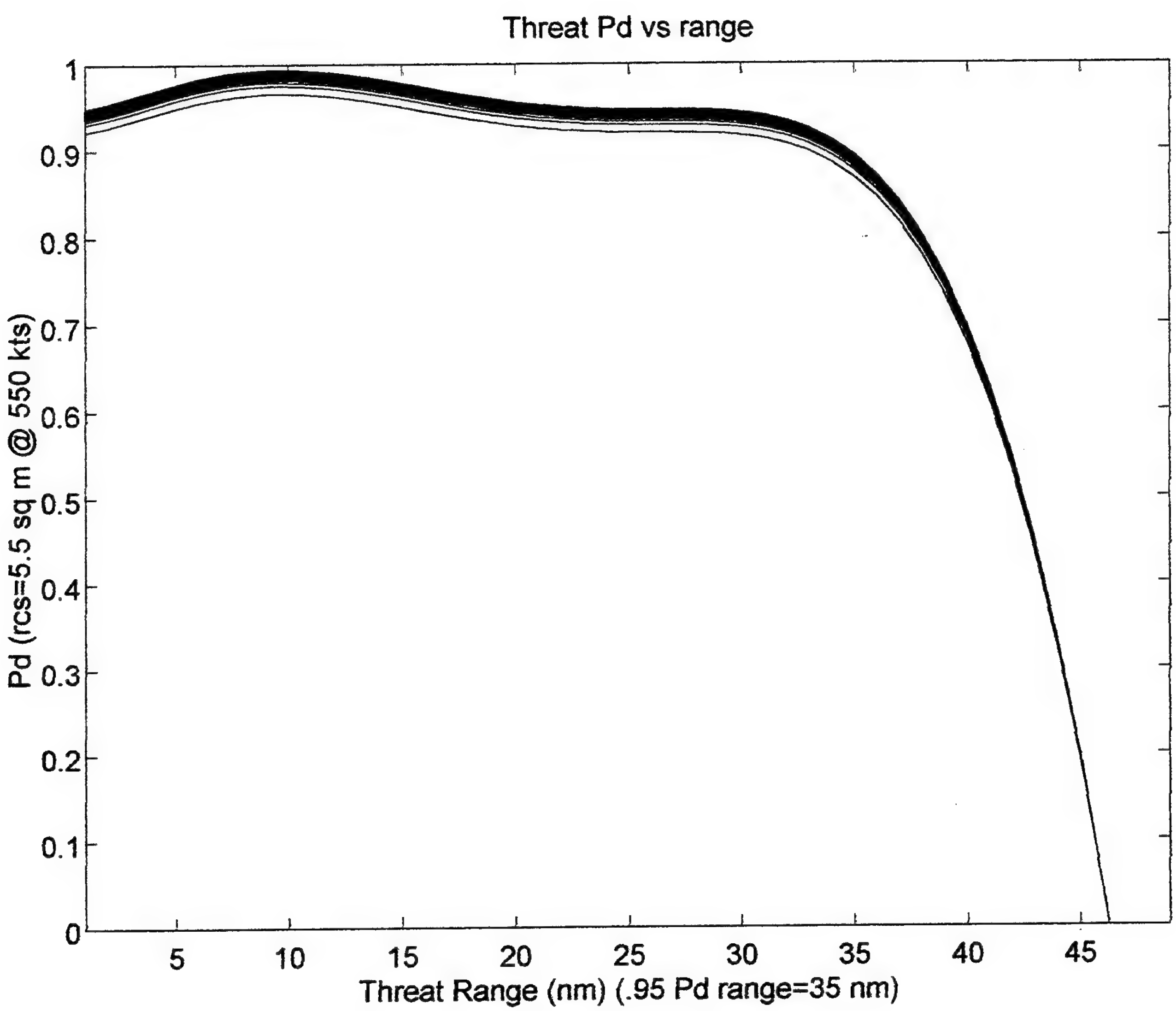

Figure (4). Detection density function plot example. Altitudes for each curve use a 10-foot increment. The lowest-valued curve corresponds to a threat at a 10-foot altitude. The highest-valued curve is for a threat at 1000 feet. Detections are horizon-limited before this function is evaluated, preventing over-the-horizon detections.

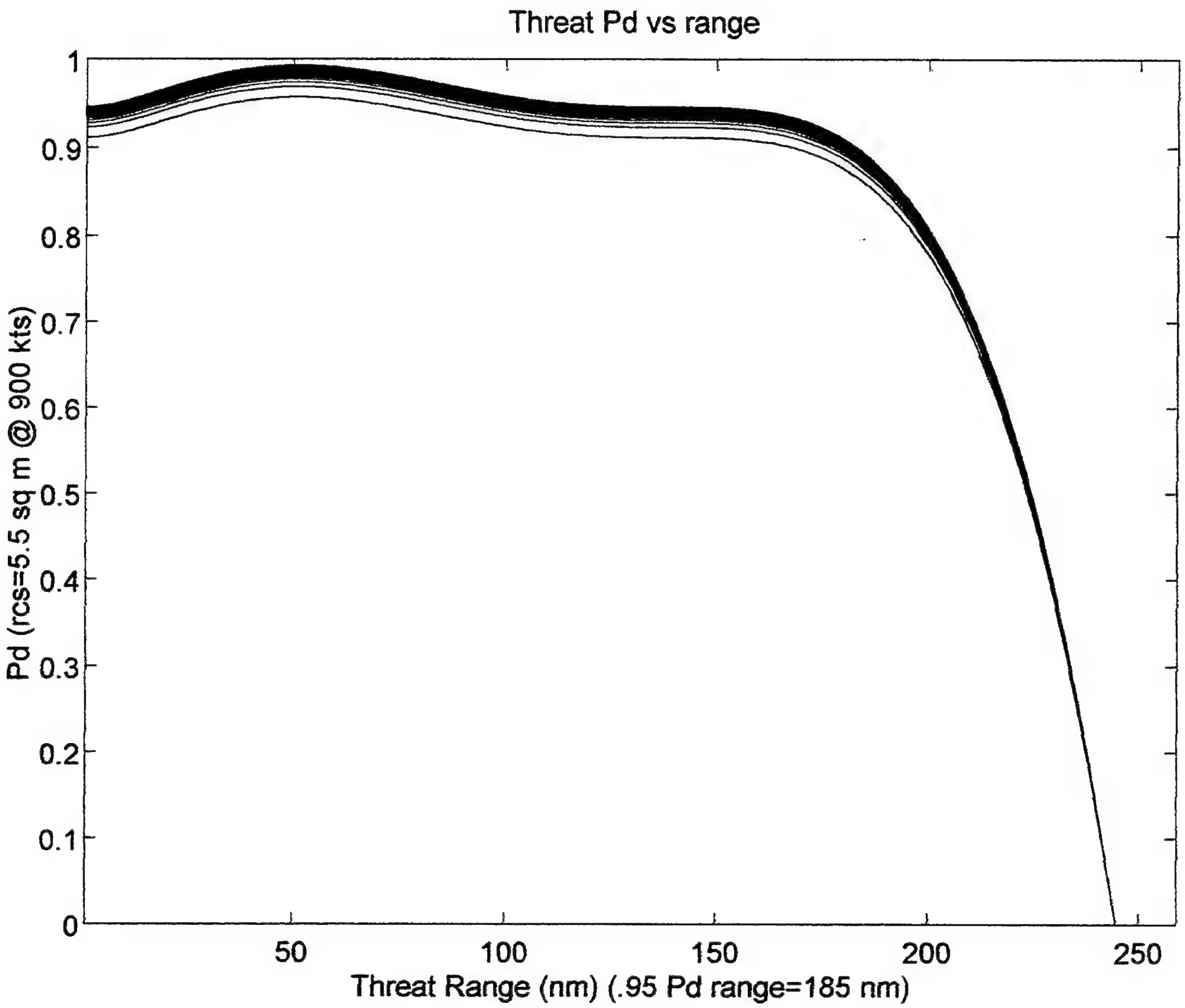

Figure (5). Detection density function plot example. Altitudes for each curve use a 10-foot increment. The lowest-valued curve corresponds to a threat at a 10-foot altitude. The highest-valued curve is for a threat at 1000 feet. Detections are horizon-limited before this function is evaluated, preventing over-the-horizon detections.

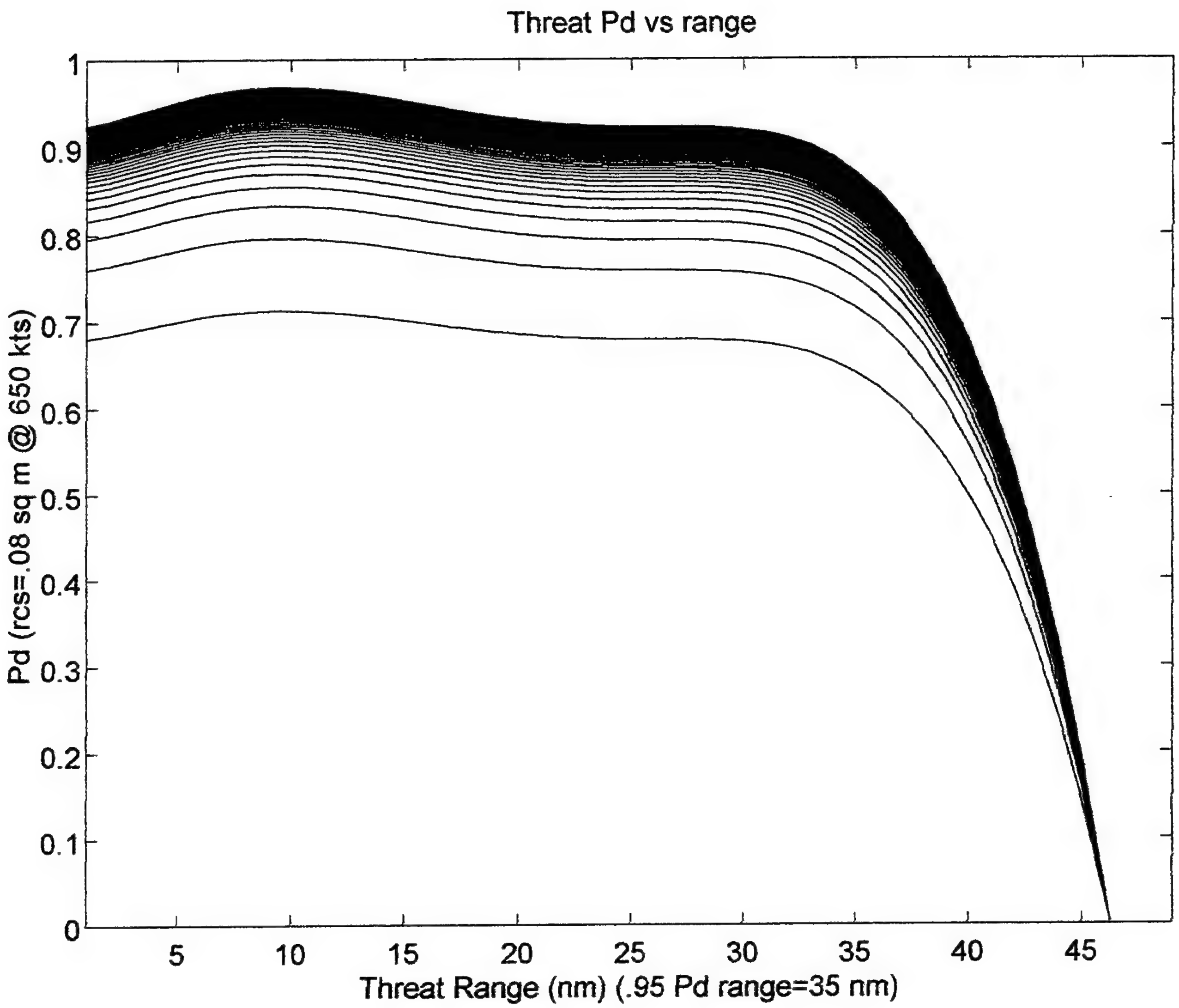

Figure (6). Detection density function plot example. Altitudes for each curve use a 10-foot increment. The lowest-valued curve corresponds to a threat at a 10-foot altitude. The highest-valued curve is for a threat at 1000 feet. Detections are horizon-limited before this function is evaluated, preventing over-the-horizon detections.

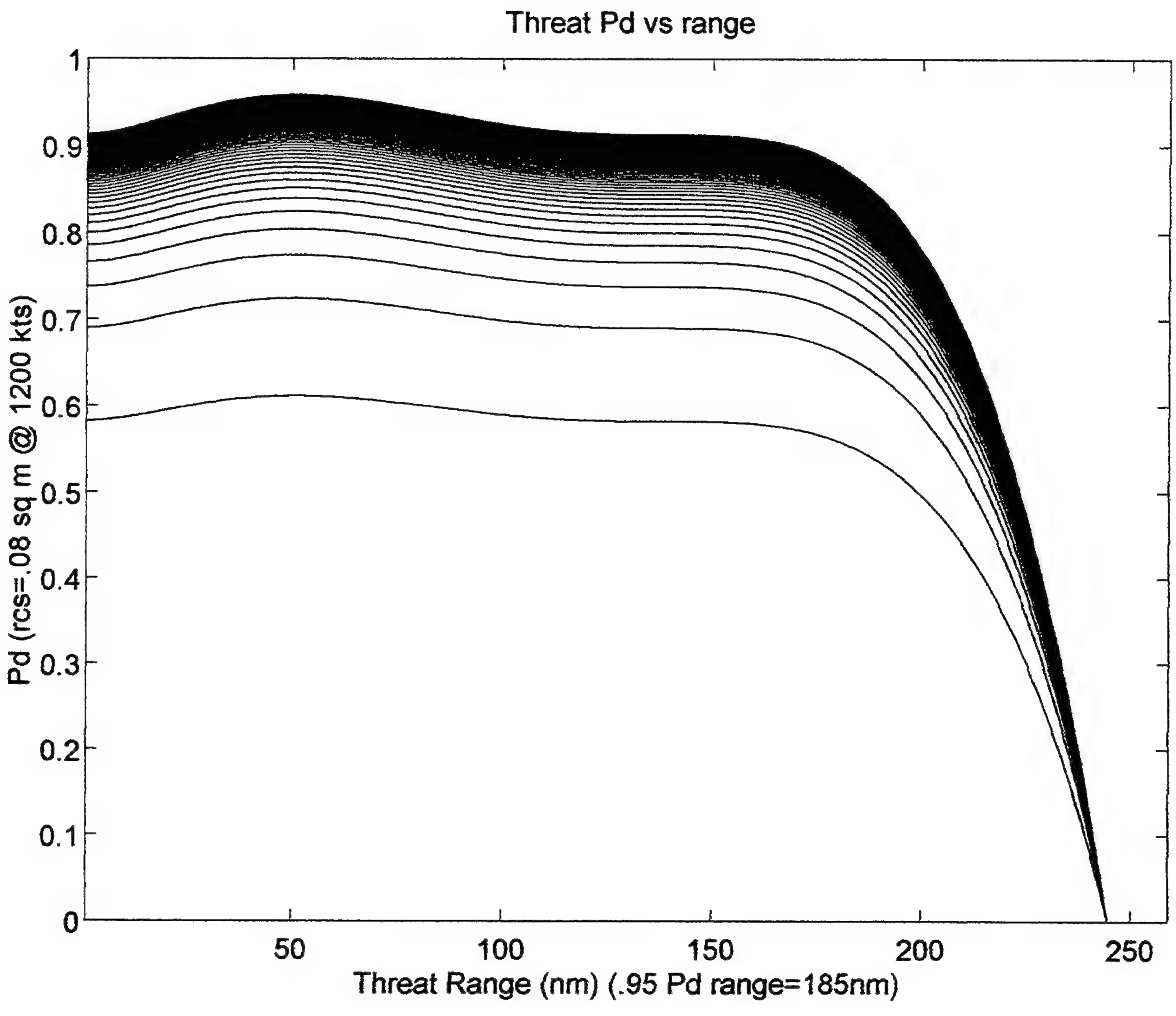

Figure (7). Detection density function plot example. Altitudes for each curve use a 10-foot increment. The lowest-valued curve corresponds to a threat at a 10-foot altitude. The highest-valued curve is for a threat at 1000 feet. Detections are horizon-limited before this function is evaluated, preventing over-the-horizon detections.

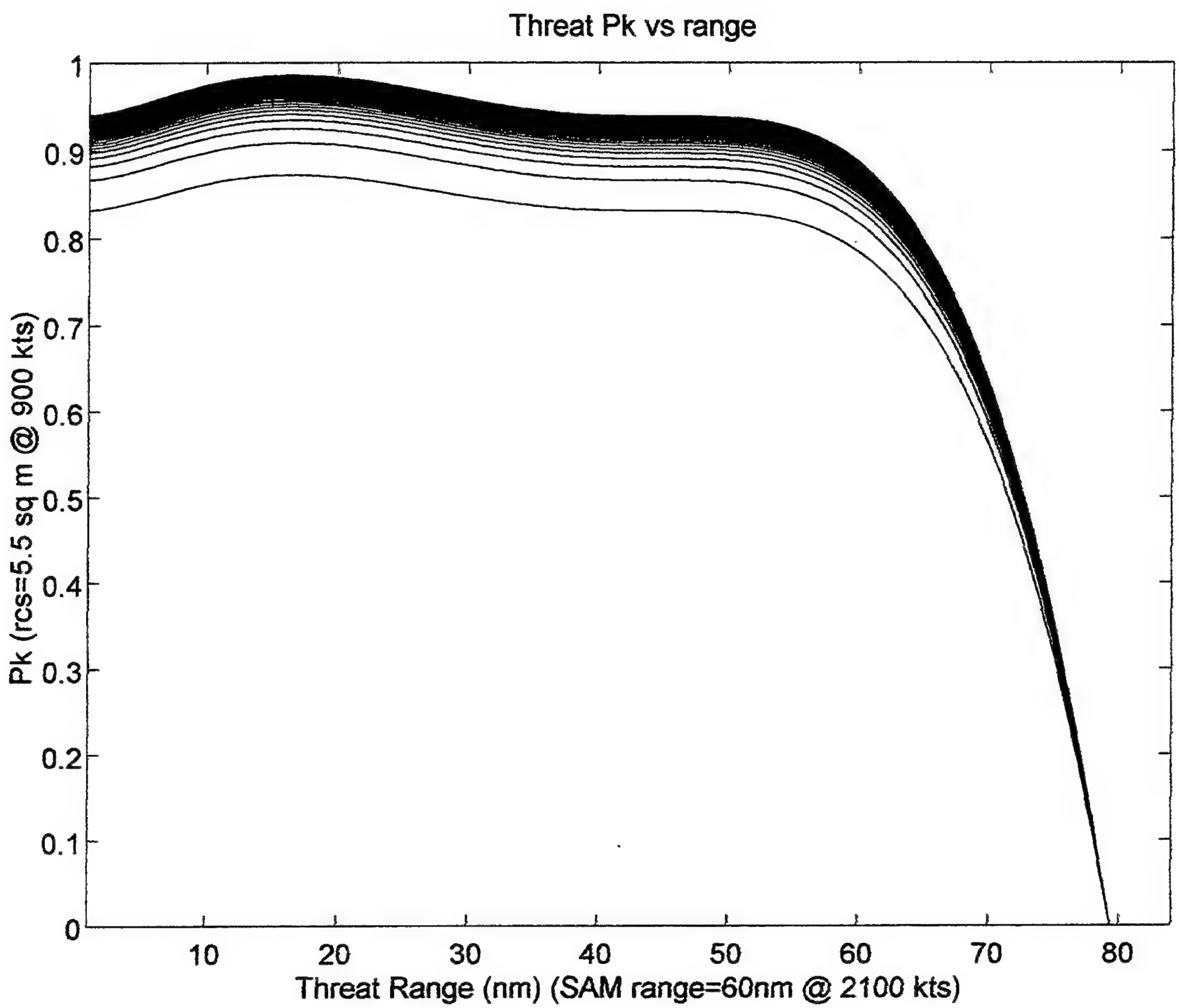

Figure (8). Kill density function plot example. Altitudes for each curve use a 10-foot increment. The lowest-valued curve corresponds to a threat at a 10-foot altitude. The highest-valued curve is for a threat at 1000 feet. Kills cannot be made beyond the sensor horizon, which is not shown in this plot.

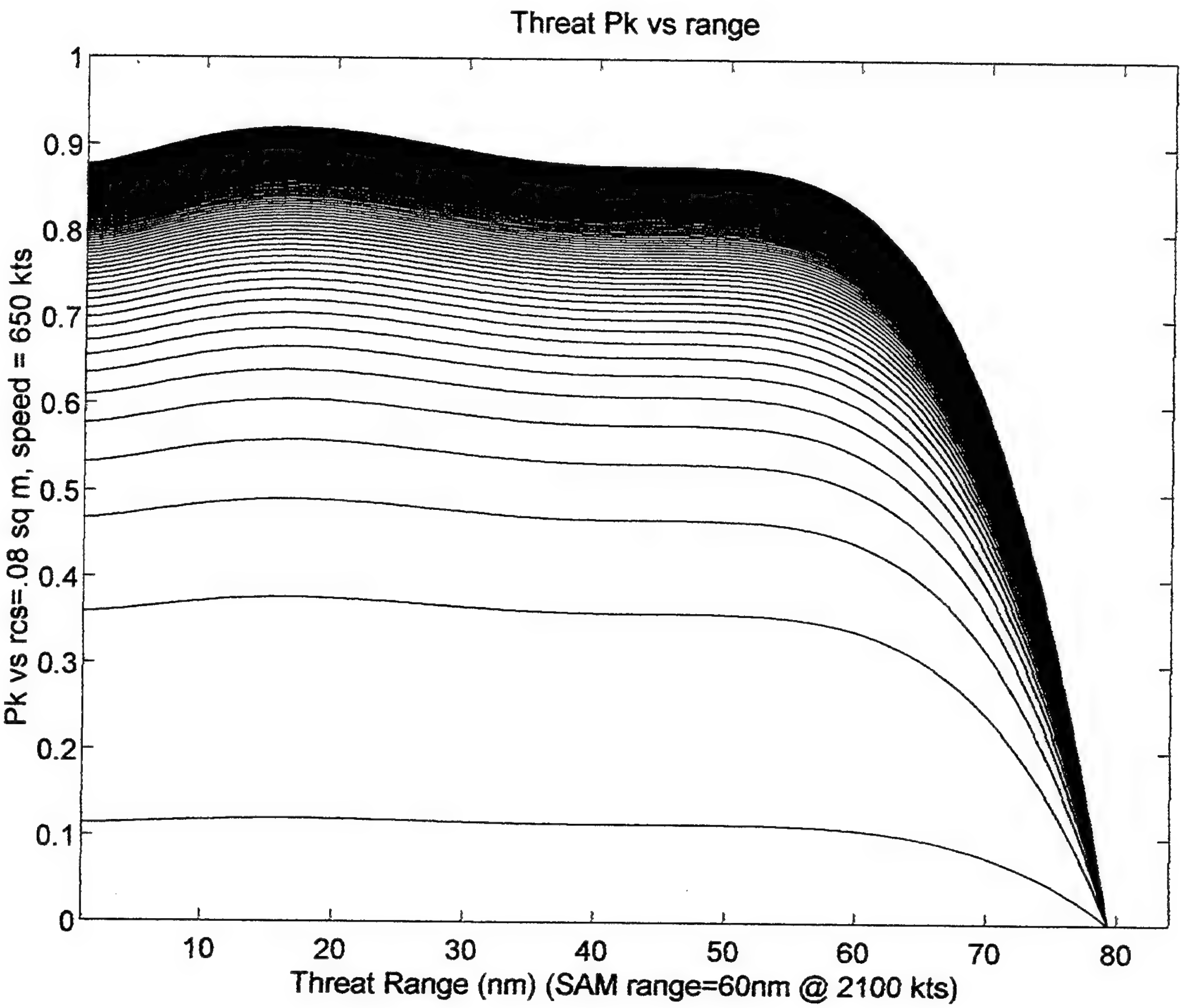

Figure (9). Kill density function plot example. Altitudes for each curve use a 10-foot increment. The lowest-valued curve corresponds to a threat at a 10-foot altitude. The highest-valued curve is for a threat at 1000 feet. Kills cannot be made beyond the sensor horizon, which is not shown in this plot.

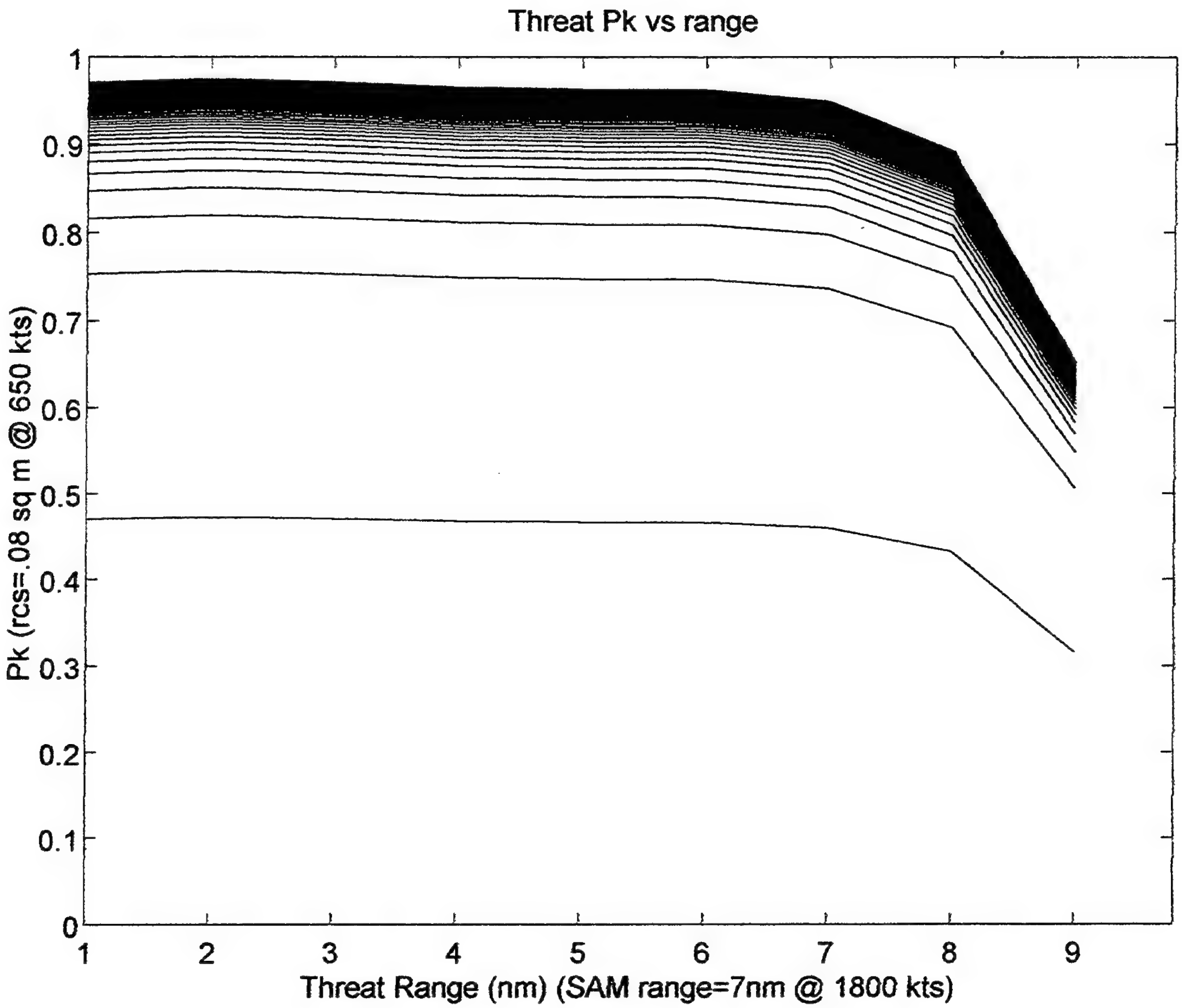

Figure (10). Kill density function plot example. Altitudes for each curve use a 10-foot increment. The lowest-valued curve corresponds to a threat at a 10-foot altitude. The highest-valued curve is for a threat at 1000 feet. Kills cannot be made beyond the sensor horizon, which is not shown in this plot.

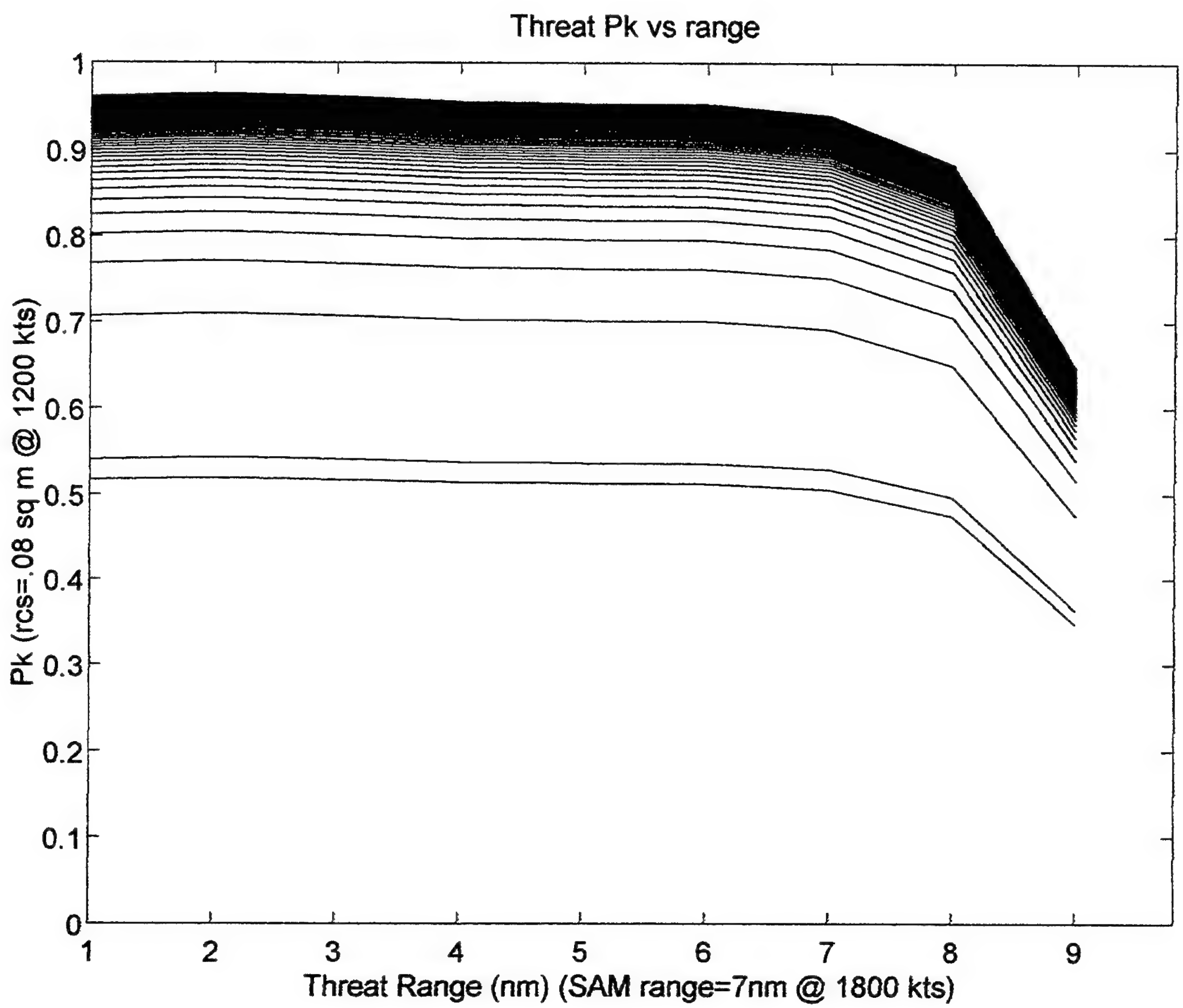

Figure (11). Kill density function plot example. Altitudes for each curve use a 10-foot increment. The lowest-valued curve corresponds to a threat at a 10-foot altitude. The highest-valued curve is for a threat at 1000 feet. Kills cannot be made beyond the sensor horizon, which is not shown in this plot.

Three random variables are created by the stochastic parameters: detection time, engagement time, and engagement outcome. They represent the main variables in a DTE sequence. We also randomize threat flight path course, using a uniform distribution with limits controlled by the parameters detailing the left and right limits of the threat sector. Before each replication, the simulation assigns a random bearing from within the sector to each threat. Our random variables are:

1. Threat approach bearing.
2. Detection times.
3. Engagement times.
4. Engagement outcomes.

Variable (1) follows a uniform distribution, with limits set by individual threat vehicle parameters. Variable (2) follows a continuous distribution with the density function given by the density function in Equation (2.5). Variable (3) is conditioned on the sequential outcomes of variable (1). Variable (4) is conditioned on the relationship created by variables (2) and (3) and the density function in equation (2.6). The sufficiency conditions for engagement time and engagement outcome determinations are given in the next section.

G. DTE SEQUENCE CONDITIONS

The random variables in the DTE sequence are heavily dependent. Each is conditioned on the existence of the prior at a minimum level of sufficiency. This creates a tier of possible conditions for each threat in the DTE sequence, following these conditions:

1. Threats detected in at least 3 out of 5 sequential detection sweeps may be engaged if a weapon is available and the threat is within the SAM envelope.
2. Weapon availability is regulated by the platform parameters MAXMIF, magazine size, and launch interval.
3. A threat must be within the SAM envelope, which has maximum and minimum range platform parameters, and a maximum engagement altitude which is calculated by multiplying the max range figure by 1000.

4. A SAM must have a P_k value of at least 0.50 at the time of engagement. This is calculated using Equation (2.5).
5. If a platform meets the requirements above, the threat is engaged. TTI is calculated using relative speed calculations for the threat and fired SAM.
6. Threats must be detectable at every sensor sweep during an engagement. If detection is lost between SAM firing and TTI, the engagement is terminated and the threat must be re-engaged subject to all earlier conditions.
7. SAMs which successfully reach TTI are evaluated for engagement outcome using the kill density function. A kill attrites the engaged threat. A no-kill requires re-engagement subject to all earlier conditions.

Each threat will be flown in the simulation until it is either destroyed or it reaches its minimum designated range from *ZZ*, the target. If the threat's minimum range has been set to a value of 0 NM, and it reaches that position, it is declared a leaker. Destroyed threats or threats completing their flight at some range greater than 0 are not counted as leakers. Each replication continues until all threats have either been destroyed or complete their flights. We repeat the process for each station until completing *m* replications. Run statistics are then calculated, and the DTE simulation begins for the next station in the list, until all stations have been tested. An output file contains the run statistics for each station. After testing every available AAW platform in every desired station, we use the stationing algorithm to generate a formation recommendation. The next section presents the mathematical formulation of the stationing algorithm.

H. SIMULATION EVENTS FLOW

Before each replication starts, the threat set is initialized; approach bearings for each threat are randomly selected from a uniform distribution; threats launched from aircraft inherit the bearing of their launch vehicle. Threats fly toward *ZZ* in our model, until either they are destroyed or reach their minimum closure range. Thus, threat courses correspond to their reciprocal bearing from *ZZ*.

Our simulation follows a time-based, event-driven format. At the beginning of each event, the clock is advanced to either the next detection sweep time or the next SAM TTI (intercept time), whichever is less. Then threat positions are updated.

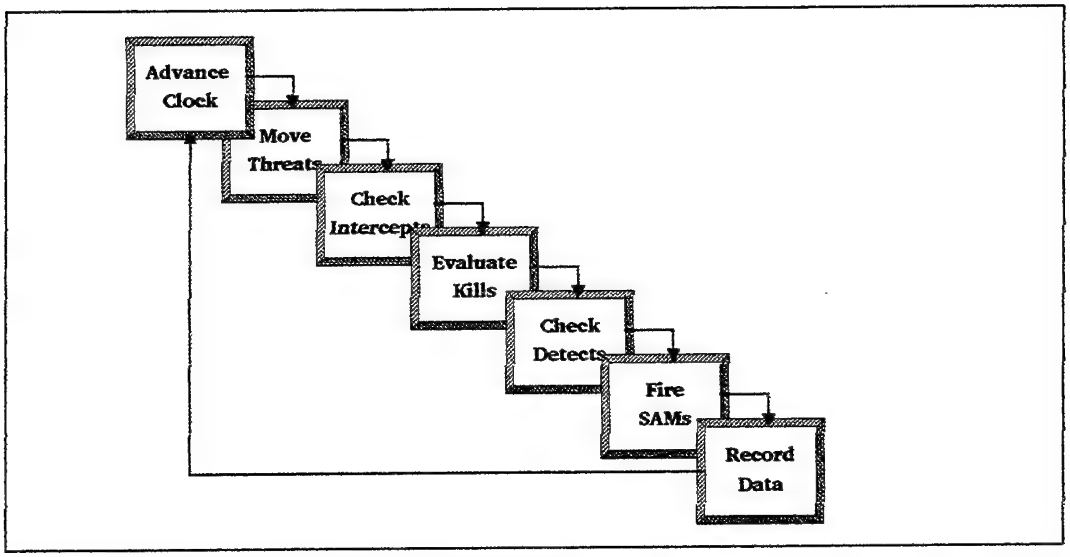

Figure (12). Simulation event flow diagram.

If the event time is a detection sweep, each threat is processed by the detection function, and its visibility condition is updated. If sufficient detections exist, the threat is not already engaged, and the engagement conditions are met, a SAM is fired, and its TTI is calculated. That ends a detection event. The clock is advanced to the next detection sweep or TTI, as appropriate.

If the next time is an intercept, the intercept is evaluated using the parametric P_k function. A requirement for detection at every sweep during an engagement adds fidelity to the model. In real-world AAW, loss of detection requires engagement termination. If the continuous detection conditions have been met, the kill is evaluated. A kill attrites the relevant threat. No-kill situations require another engagement, with the same conditions as before. That ends a TTI event.

The clock is advanced to the next event, continuing until either all threats have been destroyed or declared leakers. When that occurs, the replication ends. Run statistics are calculated and stored at that point. Threats are reset to their starting conditions, new random bearings are assigned, and the next replication begins. Our stopping rule uses a fixed number of replications. After completing the designated number of replications in a station, the platform moves to the next station and begins the process again. Figure (12) contains a flow diagram of the event cycle.

I. STATIONING ALGORITHM FORMULATION

1. Operating Parameters

The stationing algorithm minimizes the expected number of leakers by assigning AAW platforms to the stations where they are shown to be most effective by the individual platform simulation expected leaker results. Assuming *n* platforms are available, solving the complete stationing problem for optimality would require a non-linear, stochastic model with strong interaction effects. Since speed is important, a method which solves *n* individual minimizations using linking constraints was developed. For example, since no two platforms can occupy the same station, an assignment constraint must be included. Thus, at least one platform in a group will probably be located in a less-preferred location. Also, platforms might be limited in the locations where they may be stationed. That requires an availability constraint. Other operational considerations will further constrain the possible solutions. For example, there may exist minimum and maximum platform separation requirements, or a requirement to maintain a presence in specific quadrants surrounding the protected area. The model developed here includes only the assignment and availability constraints.

The *n*-variable stationing problem can be formulated as a linear programming assignment problem. That model can be translated into a network flow min-cost model; we use a relaxation algorithm on the network model to give a faster solution.

2. Assignment Model Formulation

Platforms are referred to by an index *i*. Stations are referred to by their bearing *r* and distance *d* from *ZZ*. A linear programming assignment model for the stationing problem using independent AAW effectiveness data for each platform follows:

Indices:

i	Platform number 1, ..., n
r	Station radial 1, ..., 360 (degrees)
d	Station distance 0, ..., 50 (NM, arbitrary maximum)

Data:

p_{idr}	Expected number of leakers for platform i in station d,r

Variables:

x_{idr}	Binary decision for placement of platform i in station d,r

Formulation:

$$MIN \quad \sum_d \sum_r \sum_i x_{idr} p_{idr} \tag{2.6}$$

$$subject \quad to:$$

$$\sum_d \sum_r x_{idr} = 1 \quad \forall i \tag{2.7}$$

$$\sum_d \sum_r \sum_i x_{idr} = n \tag{2.8}$$

$$\sum_i x_{idr} = 1 \quad \forall d,r \quad combinations \quad allowed \tag{2.9}$$

The objective function (2.6) minimizes the total expected number of leakers. Constraint (2.7) ensures that each platform can only be placed in one location. Constraint (2.8) ensures that every platform is assigned a station. Constraint (2.9) allows only one platform per station. This formulation does not include spacing constraints for minimum or maximum allowable platform separations, or quadrant covering constraints, which also might be desired.

3. Network Flow Model Translation

Linear program assignment models can be translated into network flow models. Lawler demonstrates the reduction from an assignment model to a minimum-cost network flow problem [Ref. 6]. A source node s seeks to force n integer units of flow to a sink node t, minimizing costs, which are the expected leaker values. Nodes in the left partition represent platforms. Nodes in the right partition represent ranked stations. Each platform's data in the graphical

example has been reduced to the same set of five stations, with the same ranks for each. Arcs connect platform node i to each allowable station j. Flow capacities for each arc are set at a value of one. Arc costs are the expected number of leakers for platform i in station j. Arcs connecting source and sink nodes to platform or station nodes have infinite capacity and zero cost.

Since the linear program formulation would be represented by an $n \times n$ matrix, the resulting network flow model can be solved in exactly n flow augmentations. Shortest path computations, each with complexity $O(n^2)$, can be used to develop each augmentation, giving an algorithm which is $O(n^3)$ or better in complexity [Ref. 7].

4. Stationing Algorithm Implementation

Our stationing problem is solved using a relaxation algorithm on the network translation [Ref. 7]. We solve the unconstrained problem, then enforce constraints and resolve any infeasibilities in the partial solutions. That is, each platform is first assigned to its best station. After checking for overassignments, platforms assigned to the same station will be re-assigned using a set of simple rules:

1. The platform which has the lowest expected number of leakers remains in the overassigned station. Other platforms move to their next-best ranked stations.
2. In the case of a tie, the platform whose effectiveness is least affected by moving to the next-best station will be moved to that station.
3. Remaining ties are broken randomly: One platform will remain in the overassigned station; all others move to their next-ranked station.

Figures (13) through (16) show a graphical demonstration of the method employed by the relaxation algorithm. Heavy, solid arcs represent station assignments for each platform. Lighter, dashed arcs represent unused feasible arcs. The partial solution in each case is represented by the solid arc paths.

The first step (Figure 13) shows the representative network with no constraints. The partial solution has platform 1 assigned with no conflicts. The remaining four platforms are overassigned to their station 1. The overassignments are resolved using the decision rules

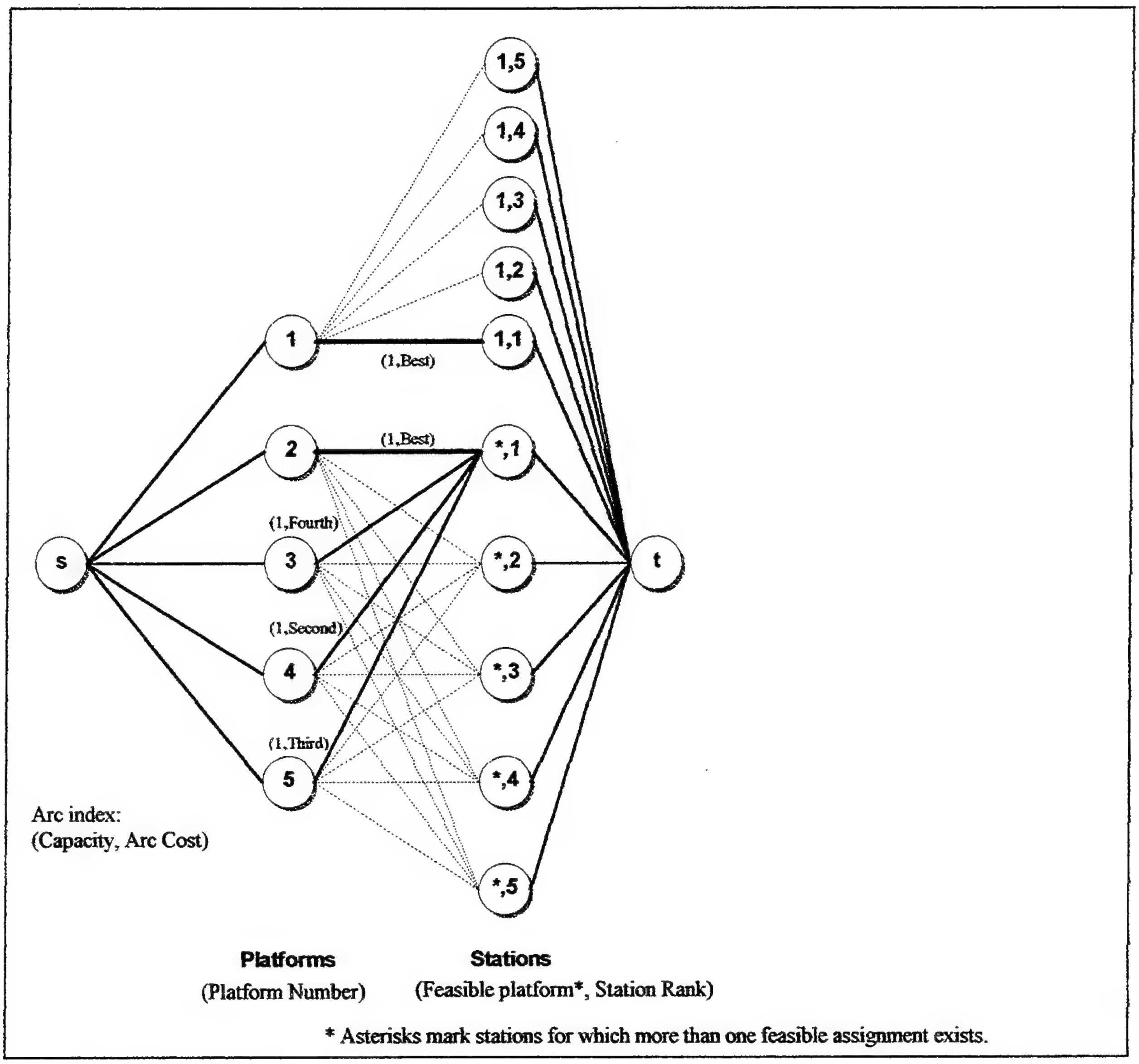

Figure (13). Graph solution process example. (Step 1 – relaxed assignment)

presented before. Platform 3 has the best performance; the remaining platforms move to their next-ranked station. The second network (Figure 14) shows the constraints enforced for the first and second platforms. The partial solution then has two platforms, numbers 1 and 2, assigned without conflict. Note that platform 1 did not enter the problem after the first step, since it was not located in an overassigned station. The remaining three platforms are overassigned to their second best station. Since platform 4 is best here, the remaining two move to the third-best station. The last two networks (Figures 15 and 16) show the steps as the last overassignments are resolved. This problem took only four iterations to solve. In fact, the design of this

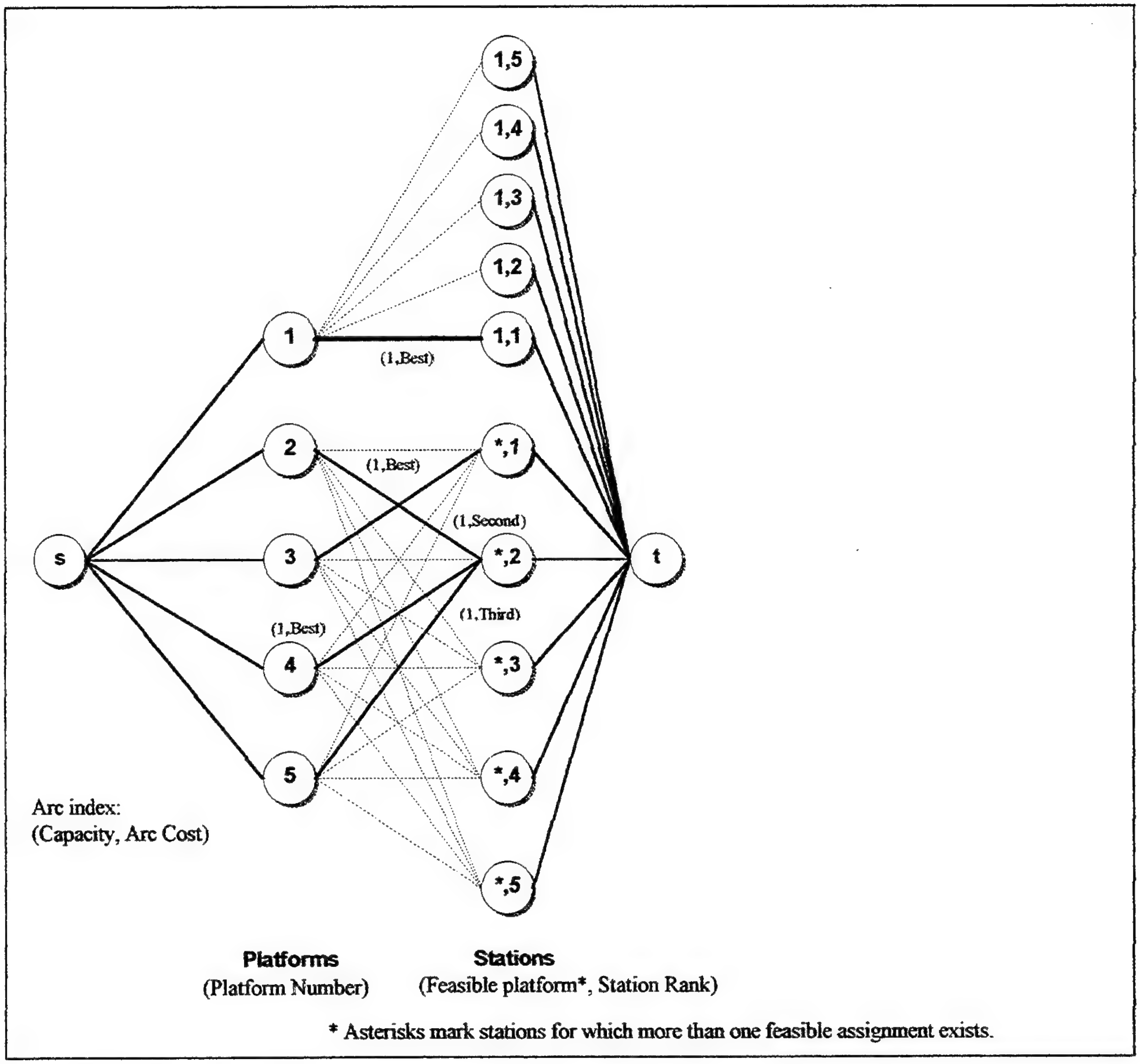

Figure (14). Graph solution process example. Second step.

algorithm allows the problem to be solved in **at most** n steps, where n is the number of platforms.

Interactions represent a large part of battle group AAW; it cannot be denied that ignoring them in the stationing process will have a negative effect. Unfortunately, including interactions leads to an extremely difficult problem to solve when speed matters. This stationing method produces a formation which provides a satisfactory level of security for protected units. Although the solution is not optimal, it provides a reasonably fast recommendation for real-world decision-makers.

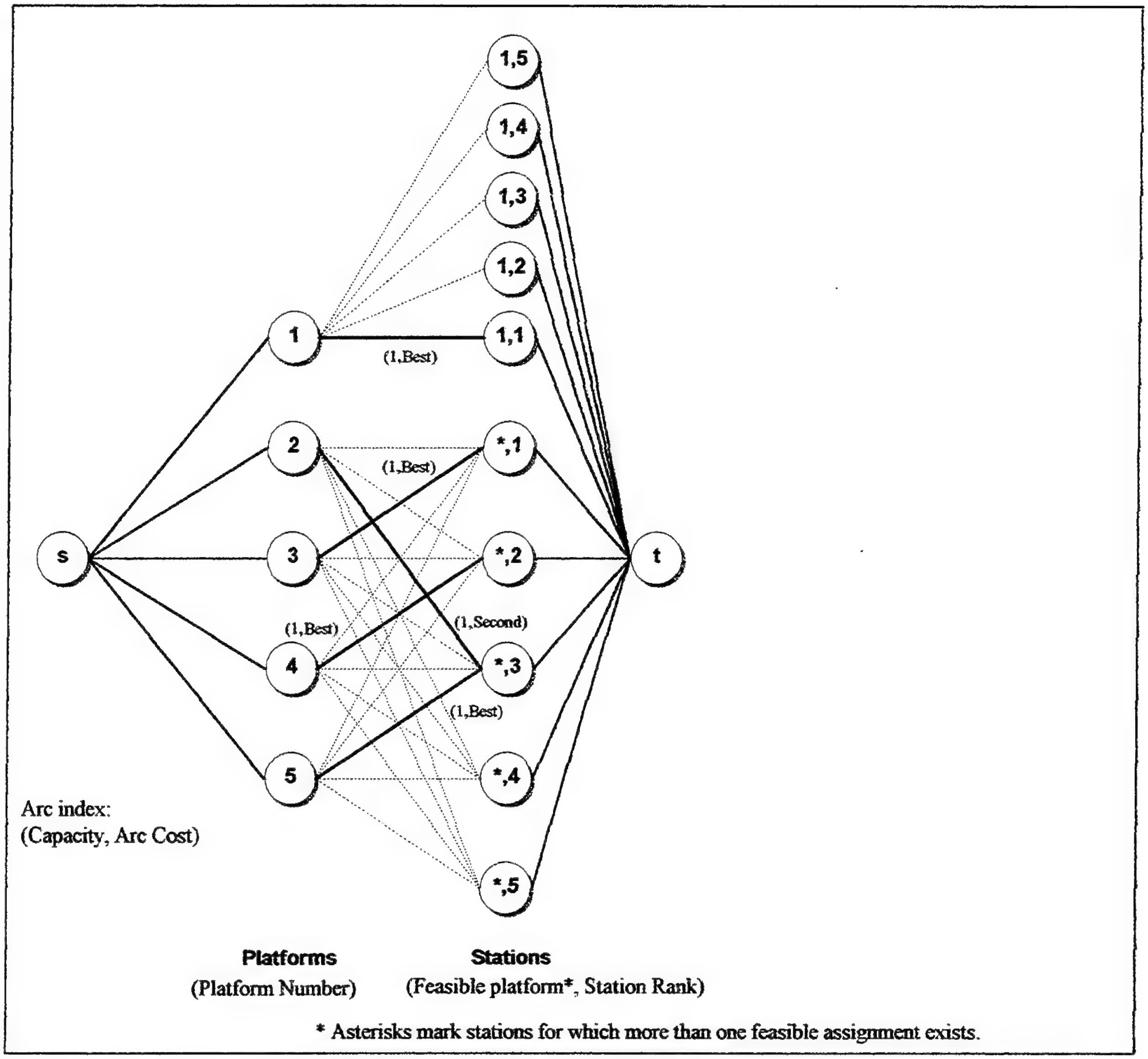

Figure (15). Graph solution process example. Third step.

In order to test the stationing algorithm solution, a method for evaluating and comparing screens is needed. For this, another simulation model can be used. Its primary features will be presented next.

J. BATTLE GROUP SIMULATION REQUIREMENTS

Testing a battle group requires a simulation which incorporates the interactions present in battle group AAW. The battle group DTE simulation allows comparison of different screens versus a constant threat set. Sensitivity analyses of changing AAW capabilities versus a

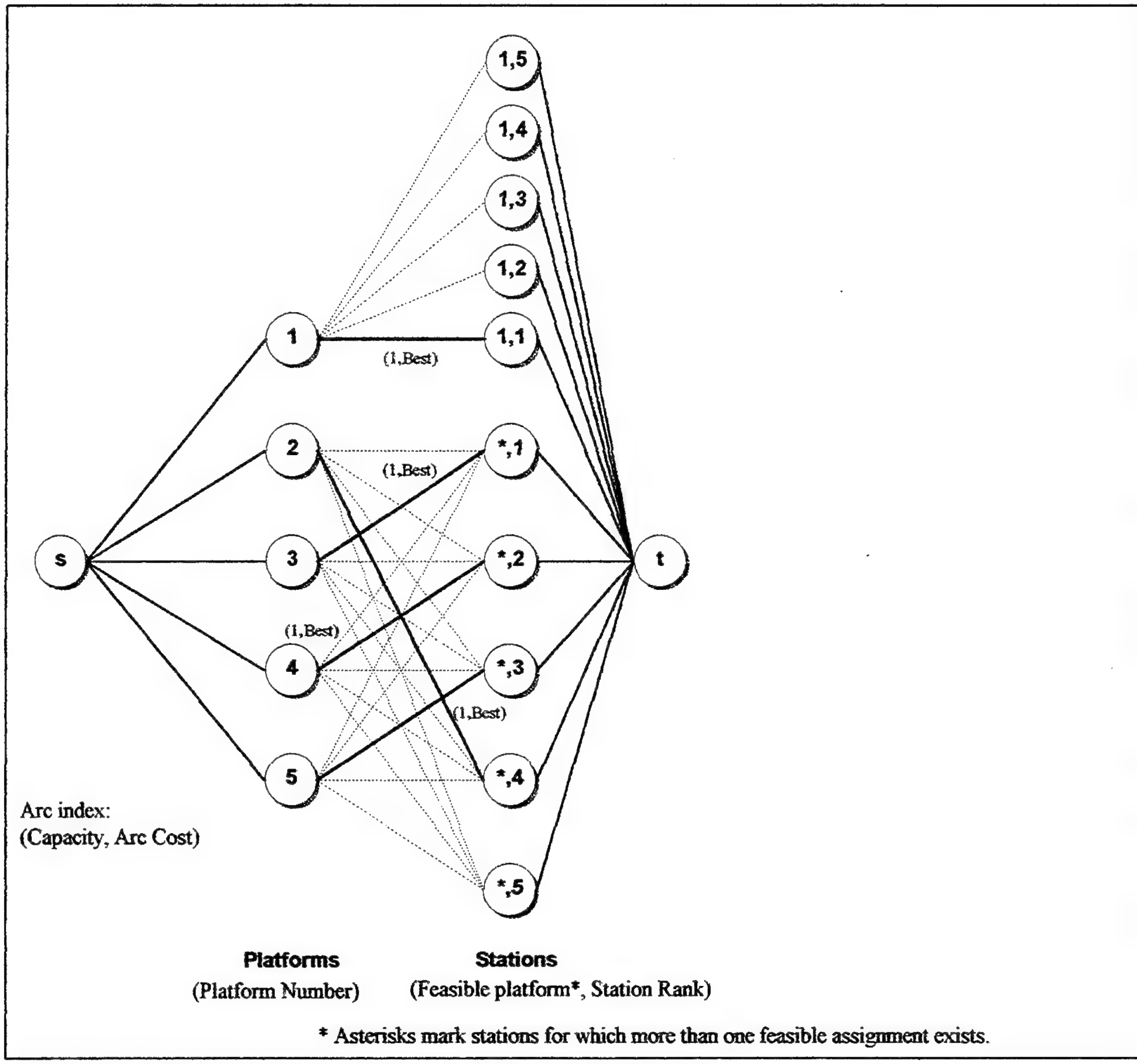

Figure (16). Graph solution process example. Final solution.

common threat set, sensitivity to changes in a threat set, or combinations of those changes can also be evaluated. Formations can also be tested without the use of the single-platform DTE simulation and the stationing algorithm, simply by creating a set of AAW platforms, a station list, and a threat set. Once all desirable formations have been tested, the decision-maker views the simulation results and compares their results to help make a formation choice.

The remaining chapters present a prototype tool developed to accompany this thesis. It uses the methodology created here, and was created to allow a concrete basis for further study.

III. PROTOTYPE INTERFACE

A. BASIC FEATURES

We developed a prototype software tool that operates in the Microsoft *Windows 3.1* environment. It is object-oriented, and was developed using Microsoft *Visual Basic 3.0*. The software was created to allow a concrete basis for review of the methodology and its applications; since the prototype was designed as a demonstration, the simulations were not validated. Some utilities programs also were developed, including graphics display options for the simulation runs. The software will operate on any computer equipped with *Windows 3.1*.

The prototype is called *BGSAMS* (Battle Group Stationing Algebraic Modeling System). It consists of five main elements: a threat database, an AAW platform database, a single-platform DTE simulation, the stationing algorithm, and a battle group DTE simulation. This chapter discusses use of the platform database, the threat database, the threat set editor, the station editor and the single-platform simulation.

B. DATA UTILITIES USE

1. Threat Database

The first step in creating an AAW test scenario is creating threat vehicles. The threat database contains threat vehicle parameters which will likely be common to similar vehicles in different scenarios. Vehicles can be any airborne threat; fighters, bombers, and missiles will comprise the most typical vehicles. Users can add, delete, or modify vehicles according to their needs. Figure (17) shows an example of a threat vehicle database record. This window can be accessed by selecting the **Edit-Threats-Individual** option in the *BGSAMS* menu. Changes are recorded as they are made, removing the need for a *save* command. Deleted records (accomplished clicking the **Delete** button) cannot be recovered; new record parameters must be manually entered after selecting the **Add Threat** option in the database window.

Parameters are listed next to their corresponding values. Units for each parameter remain consistent throughout the entire software package. For example, ranges are expressed in

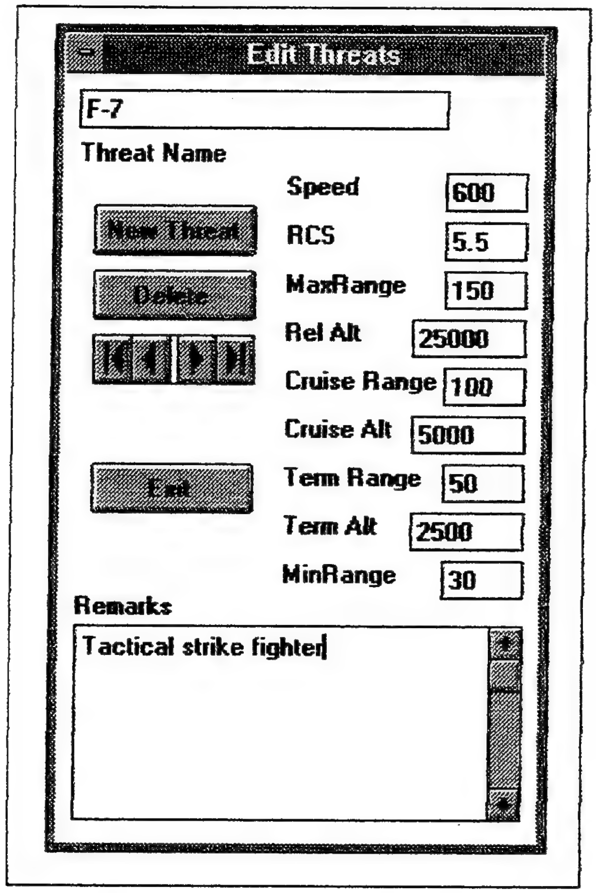

Figure (17). Threat vehicle database window.

nautical miles, bearings in degrees, altitudes in feet, speeds in knots, time in seconds, and RCS in square meters. We use nose-aspect RCS. Names of individual threats exist in conjunction with database comments to assist users in organizing their database information. Parameters, names and comments can be entered or modified by clicking on the desired text field and typing any changes. Double-clicking on a field will select all data in that field for modification.

Three independent flight stages are used for each vehicle; the threat will enter the simulation at the range entered in the *MaxRange* field. Launch time for each threat vehicle will be discussed in the next section. After launch at *Max Range*, a threat flies at *Rel Alt* (release altitude) feet until reaching the range entered in the *Cruise Range* field. At that point, the threat changes to the altitude designated in the *Cruise Alt* field. It continues at that altitude until reaching *Term Range* (terminal range), where it changes to the altitude given in *Term Alt*. After that, the threat continues until it reaches *MinRange* nautical miles from the origin. Threat do not have to close to a zero-valued *MinRange*. Also, there is no requirement for an altitude pattern

among the three legs. Each threat flies at a single speed throughout the simulation, designated by the *Speed* field. After completing all modifications or additions, click the **Exit** button, closing the individual threat vehicle editing session. Table (1) contains our fictional example threat vehicle parameters.

The next step will be to group individual vehicles into a set of threats.

Vehicle Name								
Speed	*RCS*	*Max Range*	*Release Alt*	*Cruise Range*	*Cruise Alt*	*Term Range*	*Term Alt*	*Min Range*
F-7								
550	5.5	150	15000	100	200	50	1500	35
F-7 Supersonic & Low								
900	5.5	150	2500	125	200	45	500	35
Poupon								
650	0.08	48	1500	45	100	12	20	0
Kidder								
1200	0.08	35	250	22	100	10	25	0

Table (1). Fictitious threat vehicles used in example scenario.

2. Threat Set Construction

Platforms and battle groups are always tested against a set of threats. To create or edit a threat set, select the **Edit-Threats-Group** main menu option. The window shown in Figure (18) will appear. Our prototype allows up to 75 threat vehicles in a single scenario. This limit does not depend on the number of threats flying at any given time. Users can construct as many threat sets as desired, giving each set a unique filename. Filenames follow the MS-DOS convention. Select a filename before adding threats to a list. To create a list, specify a name, click the **Open** button, then click the **Save** button. An empty file will be created, ready for entries. To modify an existing threat set, double-click its filename to load the set for editing.

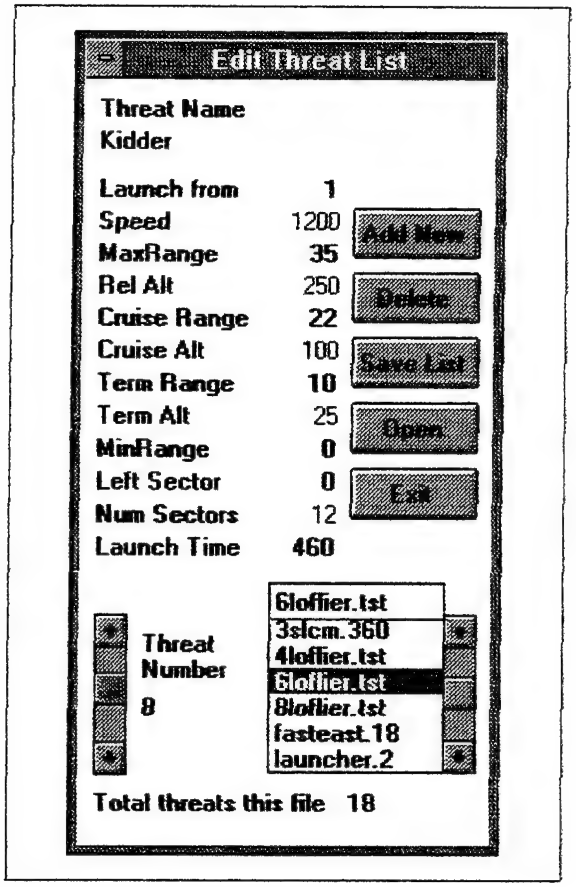

Figure (18). Threat list window.

In the set illustrated, there are a total of 10 vehicles, as shown by the text at the bottom of the window. The selected vehicle is number 8, which is how the simulation tracks the threat vehicles. Vehicle names are included only for user identification. For example, threat 8 will be launched by threat 1 at a range of 35 nm and an altitude of 250 feet. All threats proceed toward a single point, *ZZ*. When threat 8 reaches 22 nm, it will drop to 100 ft. At 10 nm from *ZZ*, it drops to 25 ft, where it remains until reaching *ZZ*. This threat can approach from any bearing within uniformly distributed threat sector of 000-359 degrees, which is designated by the *LeftSector* and *NumSectors* data fields. *LeftSector* corresponds to the first bearing in a clockwise rotation, and *NumSectors* corresponds to an even multiple of 30-degree increments for each sector size. That is, there are 12 subsectors of 30 degree radius in one rotation. A threat sector of 030-060 would have a *LeftSector* value of 30, with a *NumSectors* value of 1; a sixty-degree sector would have a *NumSectors* value of 2. If the *LeftSector* value creates a situation where a

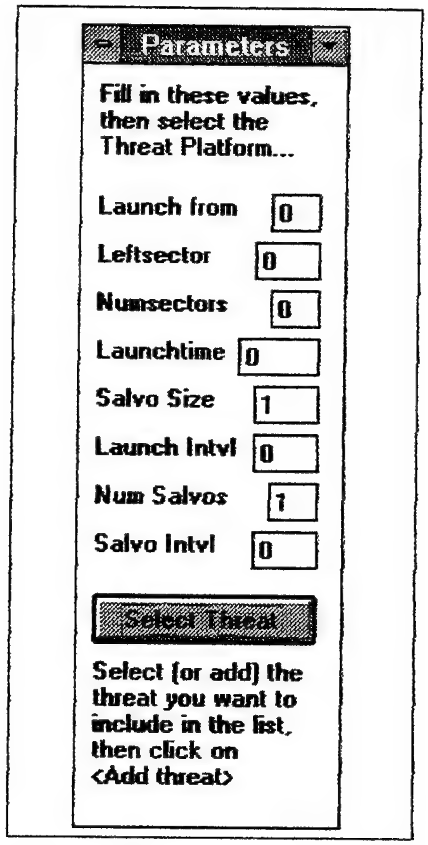

Figure (19). Threat vehicle variable parameters window.

value greater than 359 is generated, the simulation makes corrections. Therefore, a threat can use any *LeftSector* value less than 360. A value of 0 for *NumSectors* creates a deterministic, single-valued threat axis for that threat.

A connection between threat 8 and threat 1 exists; Threat 8 will use the same approach bearing as threat 1. All other parameters are independent. Launch times must be determined by the user during scenario development. We use the formula given below.

To add a threat to a list, push the **Add New** button. At that point, the variable parameters window will appear, as shown in Figure (19). Clicking the **Select Threat** button will bring up the threat database window. It will have a slight change in its control buttons; the **Delete** button will not appear, but the **Add to List** button will be visible.

Page through the threat database until the desired threat vehicle is found. We recommend adding launch platforms before their weapons, since the launch platform vehicle number is needed as a parameter for its weapons. Thus, a launch platform will already have a

vehicle designator in the threat list when its weapons are added. There is no requirement for platforms and their weapons to occupy adjacent threat list locations. New threat vehicles can also be created by pressing the **New Threat** button and entering the fixed parameters. The new threat vehicle cannot be used in a threat set until it has been appended to the database; this can be accomplished by paging forward or backward one record, followed by reselecting the new vehicle.

After selecting the desired threat vehicle, its variable parameters are entered in the parameters window. If the vehicle is a launch platform, enter a '0' value in the *Launch Platform* field. If it will be launched from another threat, enter that vehicle's list number. The values in the *Leftsector* and *Numsectors* fields determine the threat sector, as described earlier. *Launchtime* specifies the time, in seconds, from problem start (time 0) at which the vehicle will enter the simulation. The remaining fields allow the inclusion of a number of the same vehicles in one entry; *Salvo Size* determines the number fired in each of *Num Salvos* launch cycles. Launches within cycles are spaced at *Launch Intvl* seconds; launch cycles are spaced at *Salvo Intvl* seconds.

If a new threat is to be launched from another threat, we can calculate its launch time so that the fired threat enters the problem at a location near the launch vehicle. Subtract the *MaxRange* of the fired weapon from the *MaxRange* of the launch vehicle, and divide the result by the *Speed* of the launch vehicle. Multiply the resulting hour time value by 3600 to obtain a number of seconds value for the *Launchtime* field. If a salvo has more than one vehicle, the first will be launched at the *Launchtime* specified. Remaining vehicles will enter the problem at *Launchtime* plus *Salvo Intvl* times their salvo position. The next salvo begins at *Launch Intvl* seconds after the beginning of the preceding salvo.

Our example threat list has 6 fighter aircraft, each launching 2 missiles. Every threat has a 360-degree threat axis. There are 2 sub-sonic (F-7) vehicles and 4 supersonic (F-7 Supersonic & Low) fighters. All fighters enter the problem at 0 seconds. The subsonic fighters each launch 2 subsonic (Poupon) missiles, with a 10 second launch interval. The supersonic fighters each launch 2 supersonic (Kidder) missiles, with a 10 second launch interval. Using the formula for launch time, the supersonic missiles will be launched 460 at seconds. The subsonic missiles are

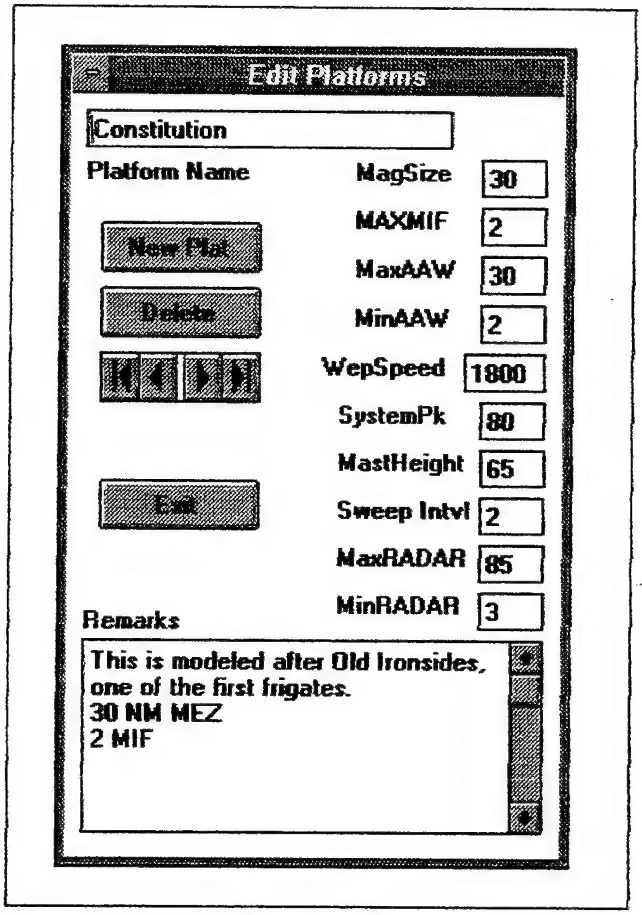

Figure (20). AAW platform database window.

launched at 650 seconds. These times will have the missiles begin their flight at the same location as their launch platforms.

3. Platform Database Use

To use an AAW platform, it must be included in the platform database. The platform database uses a format similar to the threat database; individual platforms can be created and modified as desired. Ships will each have an individual record in the database. We enter CAP in sections of 2 aircraft per platform record, since they typically operate in pairs. Since the AAW platforms do not move in our model, we gave CAP an extended *MaxAAW* range with a slower *Wepspeed* parameter, to accomodate normally moving aircraft.

BGSAMS only models one AAW weapons system for each platform. We use the longest-range, most effective weapons system. Figure (20) shows a platform database record example. Access this window by selecting **Edit-Platforms-Individual** from the main menu.

All numerical values must be integers, unless otherwise noted. *MagSize* (Magazine Size) is the number of SAMs available for firing. *MAXMIF* is the maximum number of simultaneous missiles in flight. *MaxAAW* and *MinAAW* are the respective weapon range limits. *WepSpeed* defines SAM flight speed. Since the AAW platforms do not move in this simulation, CAP should use an aggregate value of their primary weapon speed and their likely engagement speed, as discussed above. Weapon ranges for CAP also should reflect this aggregation. *SystemPk* allows a level of user control over the effectiveness of SAM systems, and should reflect relative effectiveness levels present in the mix of tested platforms. It is an integer value, and is used in P_k calculations as *SysPk* in Equation (2.5). Use integer values between 0 and 100; 0 is totally ineffective, and 100 is perfectly effective. We use values based on the capabilities against a target at an altitude where the SAM system of interest is most effective. *MastHeight* represents the RADAR antenna height; for CAP, we use their on-station altitude. *SweepIntvl* controls how often detection sweeps are made for each platform, measured in seconds. Decimal values can be entered in this field. *MaxRADAR* and *MinRADAR* are the corresponding detection ranges for a 1 square-meter target's 95 percent probability of detection ranges. We use IREPS predictions of a 95% P_d for a 1 m^2 target at an average altitude which is relevant to the threat set.

Each platform must be identified by a unique name, because the stationing algorithm sorts platform data using *Platform Name*. The *Remarks* field is included for user comments, and is not otherwise used. Table (2) gives the AAW platforms used in our example, using unclassified parameter values.

Platform Name									
Max RADAR	*Min RADAR*	*Sweep Int*	*System Pk*	*Max AAW*	*Min AAW*	*Wep Speed*	*Mag Size*	*MAX MIF*	*Mast Height*
CV-41 Midway									
185	3	4	90	7	1	1250	12	3	125
DD-972 Oldendorf									
35	1	4	90	7	1	1800	16	2	55
CG-53 Mobile Bay									
185	2	3	85	60	3	2100	88	12	58
DD-991 Fife									
25	1	4	90	7	1	1500	16	2	55
FFG-38 Curts									
65	1	4	80	25	1	1800	28	1	55
CAP F-14 x 2									
150	0	4	90	35	0	950	4	3	30000

Table (2). Unclassified AAW platform parameters used in example scenario.

4. Station File Editor

Once a threat set has been built and the desired AAW platforms have been entered, lists of test stations need to be created. For this, a station list editing utility has been included. It can be accessed through the **Edit-Stations** main menu option. Figure (21) shows an example file in the station editor.

Each station consists of two numbers on a single line entry. The first number is the range, in NM, from *ZZ* to the station. The second is the bearing, in degrees. We use the entry "0 0" for *ZZ*. To create a station list, enter a file name in the corresponding text area. Then begin entering stations in the station text box, with at least one space between range and bearing numbers and a carriage return at the end of each line (without spaces after the bearing). Station lists are saved after each entry or edit, removing the need to use the **Save** button. Follow the last station in a list with a single carriage return.

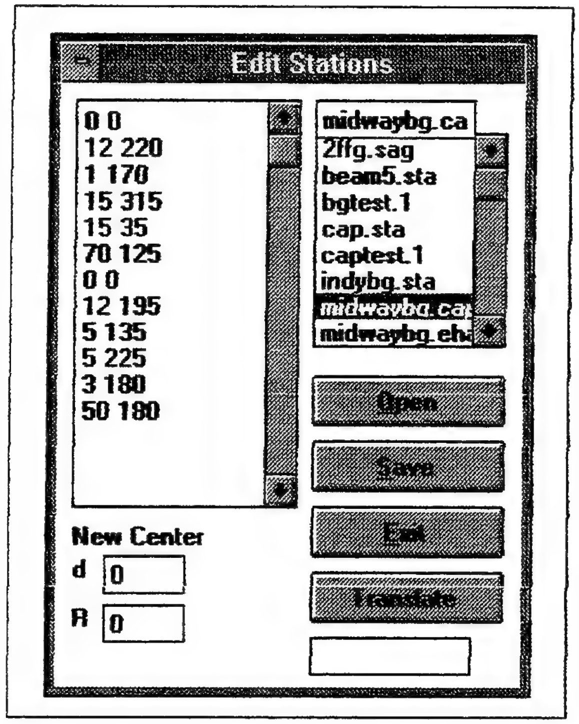

Figure (21). Station editor window.

In creating station lists, include only those which will be allowed in a formation. That may require creating an individual station list file for each platform tested. No restrictions exist as to the number of stations in each file. Platforms can be tested in any number of stations by breaking long lists into smaller segments, offering station file re-use for many platforms and shorter individual simulation runtimes. The stationing algorithm will allow the selection of multiple simulation results files for each platform. Appendix (A) contains our example station list files.

To edit a previously-created station list, select that file name from the file list. It can be double-clicked, or click the **Open** button to load the file into the station text window. Modifying the file name while there is text in the station text window will save the station text to the new file name specified. If an existing file name is double-clicked while text remains in the station text window, the station text window is cleared and loaded with the new file data. The **Translate** button and the remaining fields allow an axes translation so that threats can attack locations other than the formation center, *ZZ*, without requiring manual station recalculations. At the time this thesis was written, that algorithm did not work correctly.

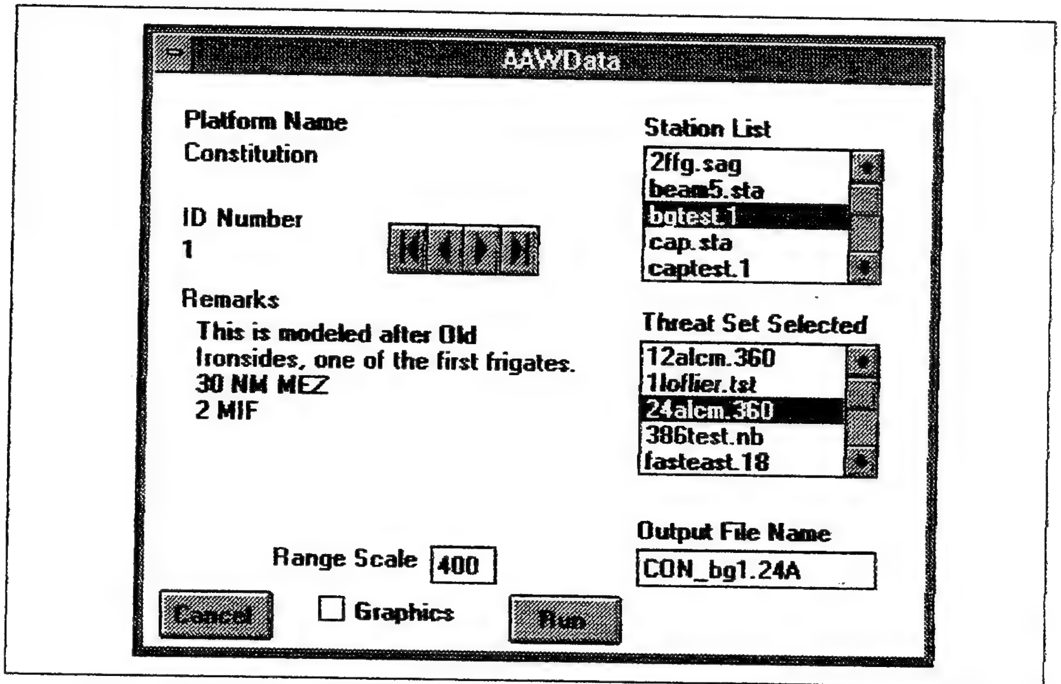

Figure (22). Single-platform simulation control window.

C. STARTING THE SINGLE-PLATFORM SIMULATION

Once a threat set has been created, and the desired AAW platforms and station lists have been added, a single-platform DTE sequence simulation can be run. We named the single-platform DTE simulation *AAWData*, since it generates data for the stationing algorithm. Its configuration window can be accessed by selecting the **Run-AAWData** menu item. Figure (22) shows the *AAWData* window.

Once the *AAWData* window appears, configuring the simulation involves selecting the AAW platform from the database, selecting the station file, selecting the threat set file, and creating a unique output file name. Duplicate file names will overwrite old files without requesting permission. We name our output files with abbreviated combinations of the platform name, station file name, and threat set name. Figure (22) gives an example.

If a graphical display of the simulation runs is desired, selecting the *Graphics* option will enable their display during the simulation runs. Using graphics will slow the simulation considerably, typically to about one-third the speed without graphics display. If graphics are displayed, the initial edge-to-edge range scale, in NM, is set by the *Range Scale* text box entry. Graphics range scales can be modified during a simulation run.

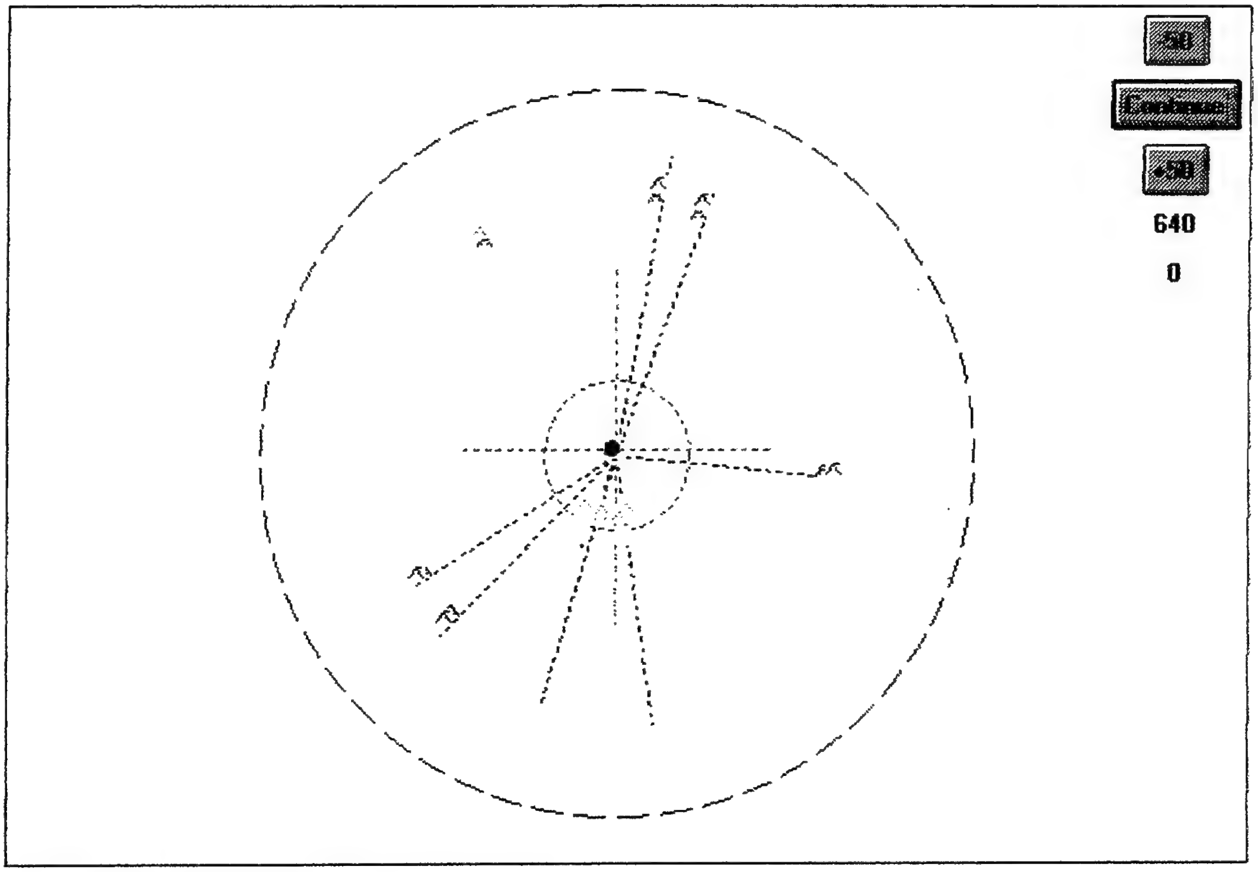

Figure (23). Simulation graphics window.

After selecting the desired configuration, start the simulation by clicking the **Run** button. The simulation graphics window will appear whether or not graphics have been selected. Graphics symbols for the simulated vehicles will only appear if the graphics option has been selected. Geographical mapping has not been included in our prototype.

During a simulation run with graphics, vehicles are represented by symbols similar to those used by NTDS (Naval Tactical Data System) consoles. Figure (23) shows an example simulation graphics window. In the simulation graphics display, symbol color denotes changes in the DTE sequence status of simulation vehicles. For example, AAW platforms are normally shown as black dots. If a platform's current missiles-in-flight equals its *MAXMIF* value, its symbol changes to a purple ring with an 'x' through the center. Symbols will change as appropriate throughout the simulation run. Also drawn at the beginning of each replication are two rings centered on the AAW platform. The red, heavy-dashed ring represents the *MAXAAW* range. The lighter, green ring represents the RADAR horizon distance for that platform versus a target at a 25-foot altitude. Crosshairs' appear at *ZZ*.

Threat vehicles appear as inverted 'v'-shaped symbols. They also change colors based on their DTE sequence status. If a threat is undetected, its symbol is grey. Once detected, it changes to purple. Symbols flashing between grey and purple have been detected by some, but not all AAW platforms. Upon engagement, a black pairing line is drawn between the threat and the engaging platform. Engaged threat symbols are blue. As threats are attrited, they disappear from the simulation.

There are controls and data displayed in the upper-right corner of the graphics window. They allow control of the range scale and pausing a simulation in progress. The **-50** button zooms the range scale in by 50 NM, measured from edge to edge. The **+50** button increases the ranges scale by 50 NM. The **Pause/Continue** button freezes a simulation in progress. The button toggles names based on the condition; a running simulation will show a **Pause** button, and a paused simulation will show a **Continue** button.

The upper number in the display shows the current time, in seconds, from the replication start. The lower number gives the current number of leakers for the replication. The number is not reset until sometime into the next replication, so that the final value for the last replication is displayed for a time after the next replication starts.

D. SIMULATION VARIABLES CONFIGURATION

The number of replications used for each station can be set using the **Edit-Preferences** main menu selection. This is how we control the quality of the expected numbers of leakers estimates. If the results produced are of comparable quality across platforms, they can be compared as a measure of effectiveness. A measure of quality is produced by the simulation, equal to the coefficient of variation multiplied by 100 and rounded to the nearest integer. This gives an easy method for tracking data quality. Smaller numbers represent higher quality; we consider any value less than or equal to 5 as acceptable quality. This corresponds to a *cv* of 0.05. Figure (24) shows a plot of the expected number of leakers and the sample standard deviations for two identical series of simulation runs, each with increasing numbers of replications. Figure (25) plots the coefficient of variation for each of the runs. *BGSAMS* produces acceptable results using at least 75 replications. Some scenarios will generate good data with fewer replications.

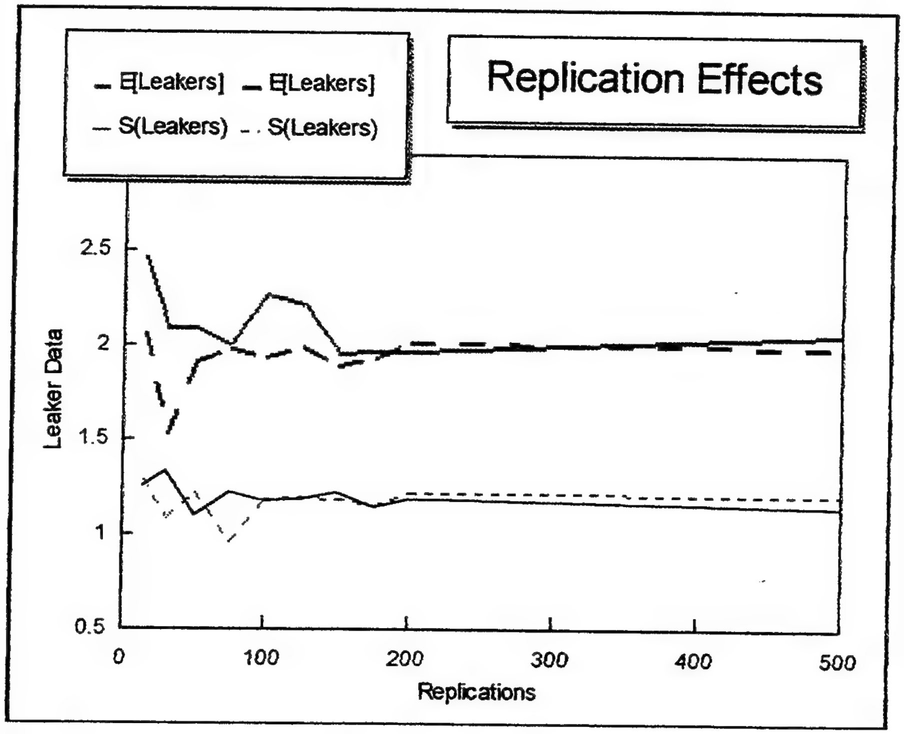

Figure (24). Replication effects on E[Leakers] and S(Leakers)

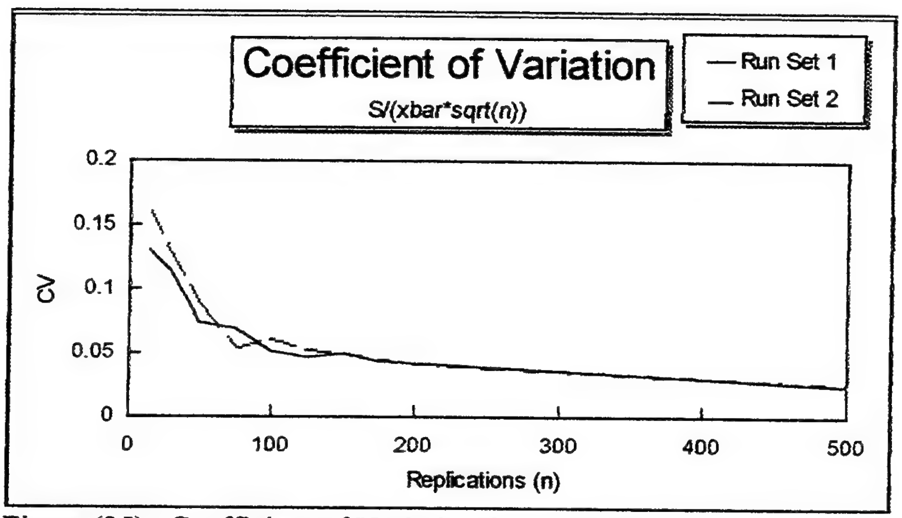

Figure (25). Coefficient of Variation for two run sequences.

Replications defaults to 1 when *BGSAMS* is first loaded. We used 100 replications for each platform in our example problem, giving a quality value of 3 or better. A quality value of 0 is suspect, since it represents a zero-variance result. Typically, that will only occur when a very small number of replications are used, or when the platform always prevents leakers.

Accessing the quality value and simulation results requires first running a simulation. To create sample data, run *AAWData* using a station list with a single station, with about 20 replications. After the run finishes, click the **Close** button in the upper-left corner of the graphics window. Then select the **Check-Sensitivity Graph** option from the main menu. The sensitivity analysis window will appear. This utility allows a text and graphical display of single-platform DTE simulation results. Figure (26) shows the sensitivity analysis window. To see the contents of a data file, single-click the file name in the file list box. The text of the data file will appear in the text box. The first two lines in the data file give the threat set file name and station list file name. Each station will have a multiple-line entry:

1. Platform name
2. Station tested
3. Probability of No Leakers * 10000
4. Quality level = *cv* * 100, rounded
5. Expected Leaker value * 1000
6. Expected number of SAMs fired, rounded
7. Expected battle length, in seconds
8. Actual runtime, in cumulative seconds from the first station start time

To display the data graphically, enter an edge-to-edge range scale. Next, click the **Show** button after single-clicking the desired data file, or double-click the data file name in the file list box. The sensitivity graph will then appear. Figure (27) shows an example sensitivity graph plot.

The sensitivity graph plots each station as a 1 NM-radius ring, with color denoting effectiveness levels for each station. The colors represent the probability of kill ranges for each

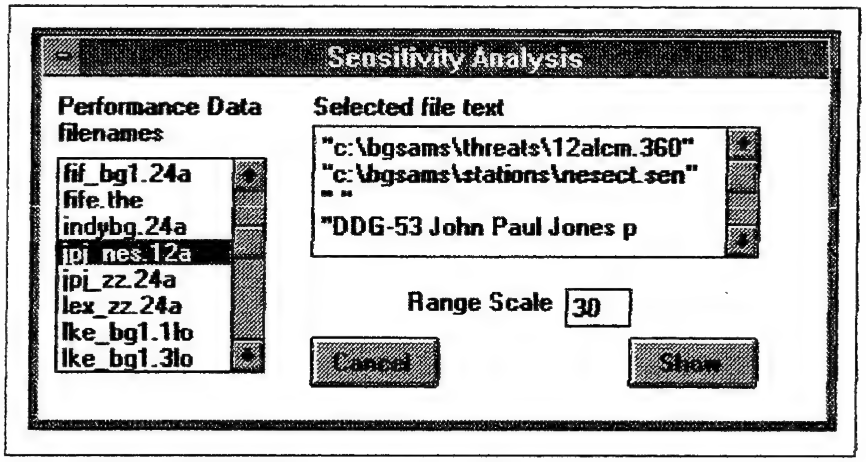

Figure (26). Sensitivity Analysis window.

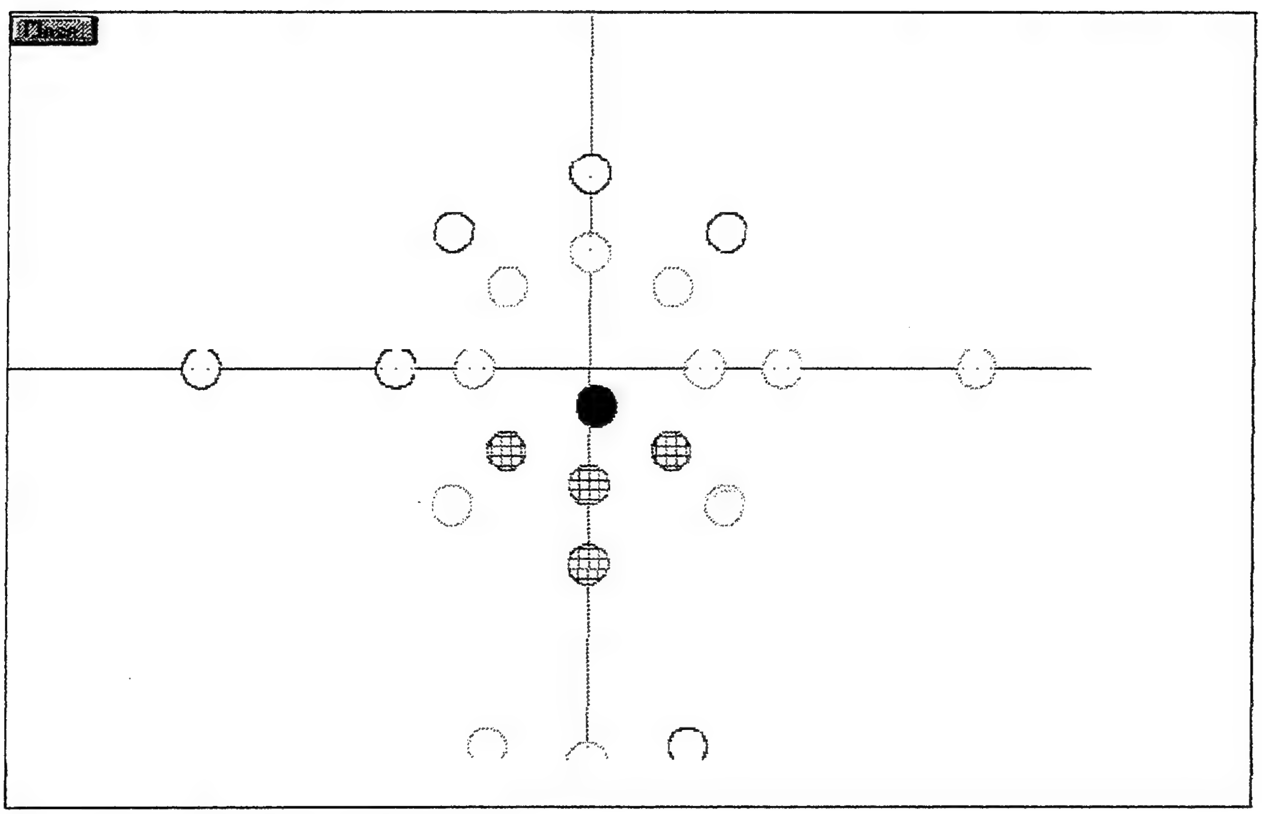

Figure (27). Sensitivity graph example.

station. Stations with associated kill probabilities greater than 95 percent will appear as black rings. Stations whose probability of kill data lies between 75 and 95 percent will appear as grey rings. Stations with a probability of kill value between 50 and 75 percent appear as yellow rings. Stations with probability of kill values below 50 percent appear as red rings. To close the graph window, click the **Close** button in the upper-left corner of the sensitivity graphics window.

To clarify the graphic shown in Figure (27), it has been modified for display without color. Stations which appear black in the actual graph are shown as a solid disc. Grey stations are shown with a cross-hatch pattern. Yellow stations are shown as light grey rings, and red stations appear as dark grey rings in the figure. The patterns do not appear in the actual graph and were added to clarify its display.

E. STATIONING ALGORITHM

Our stationing algorithm uses the relaxation algorithm described in Chapter II. To create a formation, the first step is to select the single-platform DTE simulation output files desired. The stationing algorithm sorts the data for each platform into a ranked list of stations, using the platform name included with each station's data. We included an ability to allow the selection of several output files for each platform. Figure (28) shows the stationing algorithm window.

As each file is selected, the stationing algorithm compiles a list of the 14 best stations for that platform. If fewer than 14 stations are tested, the algorithm still works, but at least the number of platforms present in the battle group must be used. The algorithm can produce a formation for up to 14 platforms, the maximum allowed in the battle group DTE simulation. Platforms with only one tested station are forced to remain in that station.

When *BGSAMS* is first loaded, the stationing algorithm retrieves the last sorted list produced from a system file. To view that list, click the **View** button in the stationing algorithm window. The sorted list contents will appear for inspection in the stationing algorithm text box. To clear all data from the list, click the **Reset** button.

Data files are sorted upon selection. If a file name is single-clicked in the file list box, its contents will appear for inspection in the text window. The text window can be scrolled to allow viewing the entire file. If the file is one that should be included, the **Sort** button can be clicked,

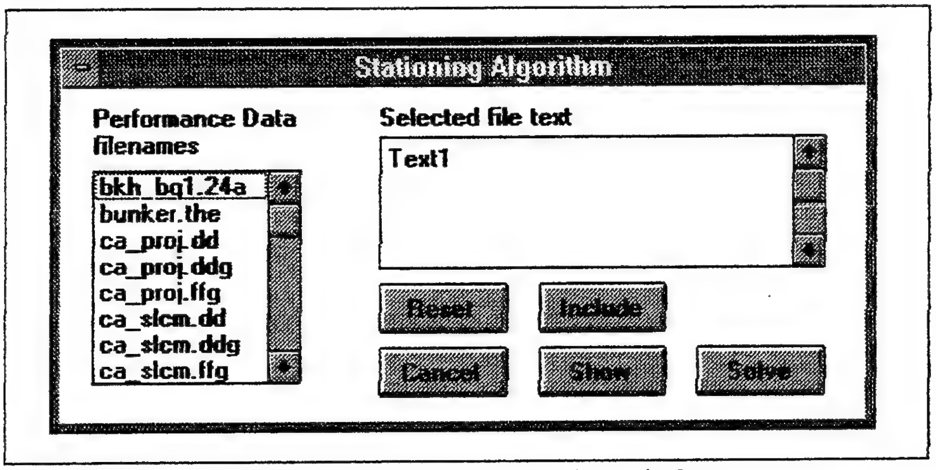

Figure (28). Stationing algorithm data selection window.

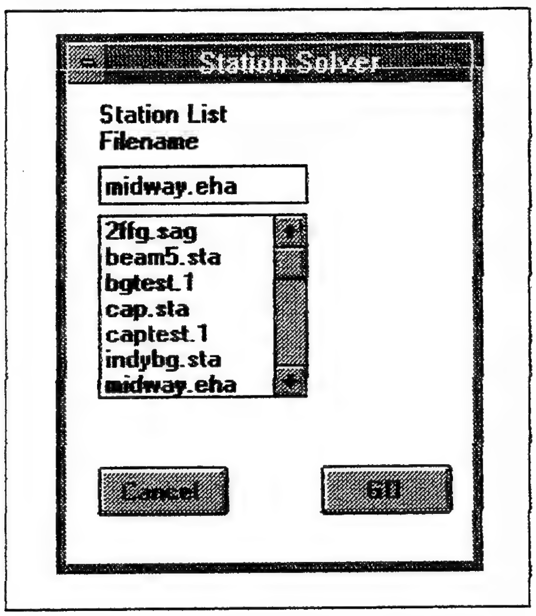

Figure (29). Stationing Algorithm solver control window.

or the filename can be double-clicked to sort the data. Data will not be included in the sorted list until this step.

Once all data files have been included in the sorted list, click the **Solve** button. The stationing algorithm solver control window, shown in Figure (29), will appear. This window controls the stationing algorithm solver and provides an output file name for the solver's formation. Specify a formation file name, then click the **Go** button.

The stationing algorithm will produce a station list which can be read by the battle group DTE simulation. Stations for each platform will be placed in the station list in the same order that platforms appear in the solver sorted data list. Because of this, it is important to be aware of the order of the AAW platforms in the battle group platform list. Creation of a battle group platform list will be presented next.

F. BATTLE GROUP DTE SIMULATION

1. AAW Platform List Construction

To test a battle group using the battle group DTE simulation, a list of AAW platforms analogous to a threat set must be created. We included a battle group editing utility for this purpose. It allows the inclusion of AAW platforms selected from the platform database in a battle group platform file. To access the battle group editor, select the **Edit-Platforms-Group** option from the main menu. The platform group editing window, shown in Figure (30), will appear.

To edit an existing battle group, select its file name and click the **Open** button or double-click the file name. The battle group will be loaded into the editor. Adding or deleting platforms follows the same format as the threat set editor; clicking the **Add New** button loads the platform database window, where platforms may be selected and edited as in the threat set editor.

A maximum of 14 platforms may be included in a battle group. Each record in the platform database counts as a single-platform. To add a platform to a battle group, select its record in the platform database and click the **Add to List** button in the platform database window. The platform will be added to the battle group.

To delete a platform from a battle group, select it in the battle group editor window, and click the **Delete** button. The platform will be deleted from the battle group. Battle group lists are not automatically saved by the battle group editor.

To create a new battle group, type its file name in the file name text box, then click the **Open** button followed by the **Save** button. An empty file will be created; save the list again after

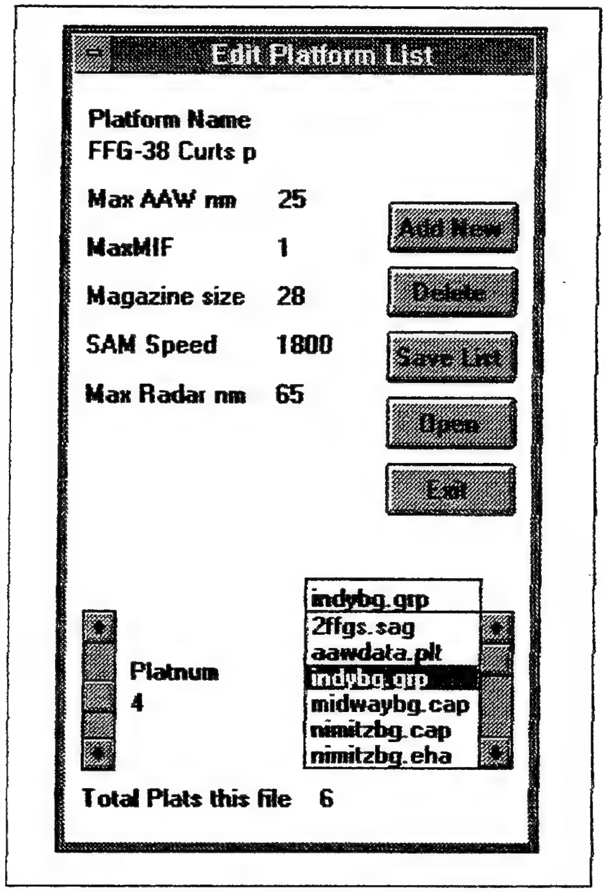

Figure (30). Battle Group editor window.

adding the desired platforms. We include the CV first in our lists, and CAP are the last plaftforms added to our battle groups.

2. Station Lists for Battlegroups

The battle group DTE simulation uses the same station list editor as the single-platform DTE simulation. Stations are assigned to each platform in matching order. That is, the first station in a station list will be assigned to the first platform in a battle group list, the second station to the second platform, and so on. If there are extra stations in the list, two possibilities may occur. If there are at least as many extra stations as battle group platforms, then the battle group DTE simulation will run simulations using each formation in sequence. The simulation will load sequences of formations until all stations have been used, or not enough remain to complete a battle group station assignment. This allows more than one formation to be tested in

a single run, a method analogous to testing multiple platform stations in the single-platform DTE simulation.

3. Stationing Algorithm and Station Lists

The stationing algorithm station lists are produced in the order that platforms are included in the stationing algorithm data sorter. We always test the CV at *ZZ* and no other stations. This forces the stationing algorithm to place the CV at *ZZ*. Other platforms are tested in any desired stations. When selecting data for the stationing algorithm, we include the CV data first, since the CV always appears first in our battle group lists. The remaining platforms' data are included in the same order that they appear in our battle group list. Using this method allows the battle group DTE simulation to read the output file of the stationing algorithm with no modifications. If platforms do not correspond to matching orders in the station and battle group lists, stations will be assigned to the wrong platforms.

G. BATTLE GROUP DTE SIMULATION CONFIGURATION

We named our battle group DTE simulation *AAWSim*. To test a battle group using the battle group DTE simulation, select the **Run-*AAWSim*** option from the main menu. The window shown in Figure (31) will appear.

Select a battle group file , station file, and threat file from the corresponding file name lists. Specify an output file and select the desired graphics options. After these actions are complete, click the **Run** button. The number of replications used will correspond to the number set in the **Edit-Preferences** window. The battle group DTE simulation graphics use the same symbols as the single-platform DTE simulation graphics. Figure (32) illustrates a battle group DTE simulation graphics output example. Its range scale has been set to 175 NM.

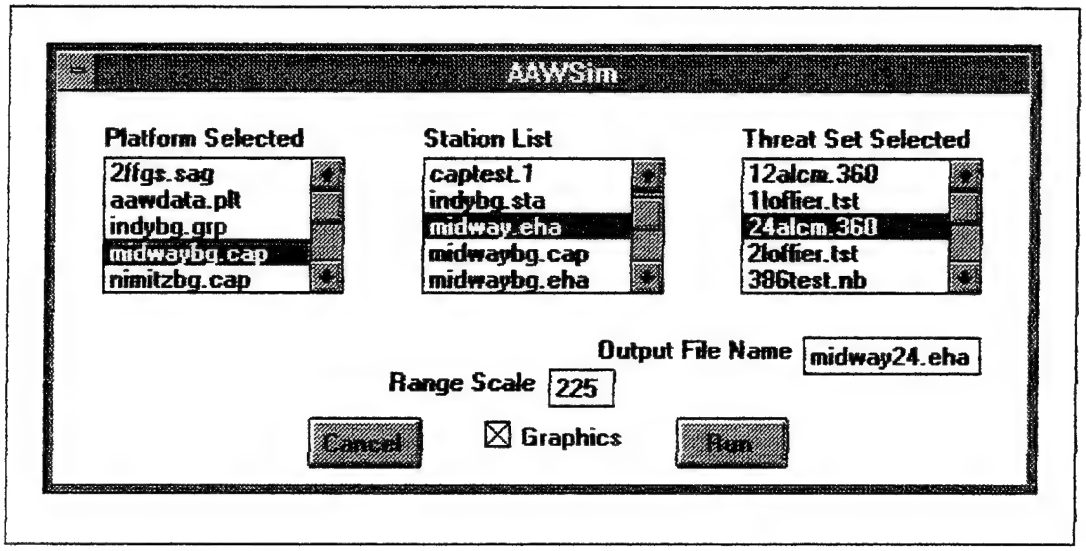

Figure (31). Battle Group simulation configuration window.

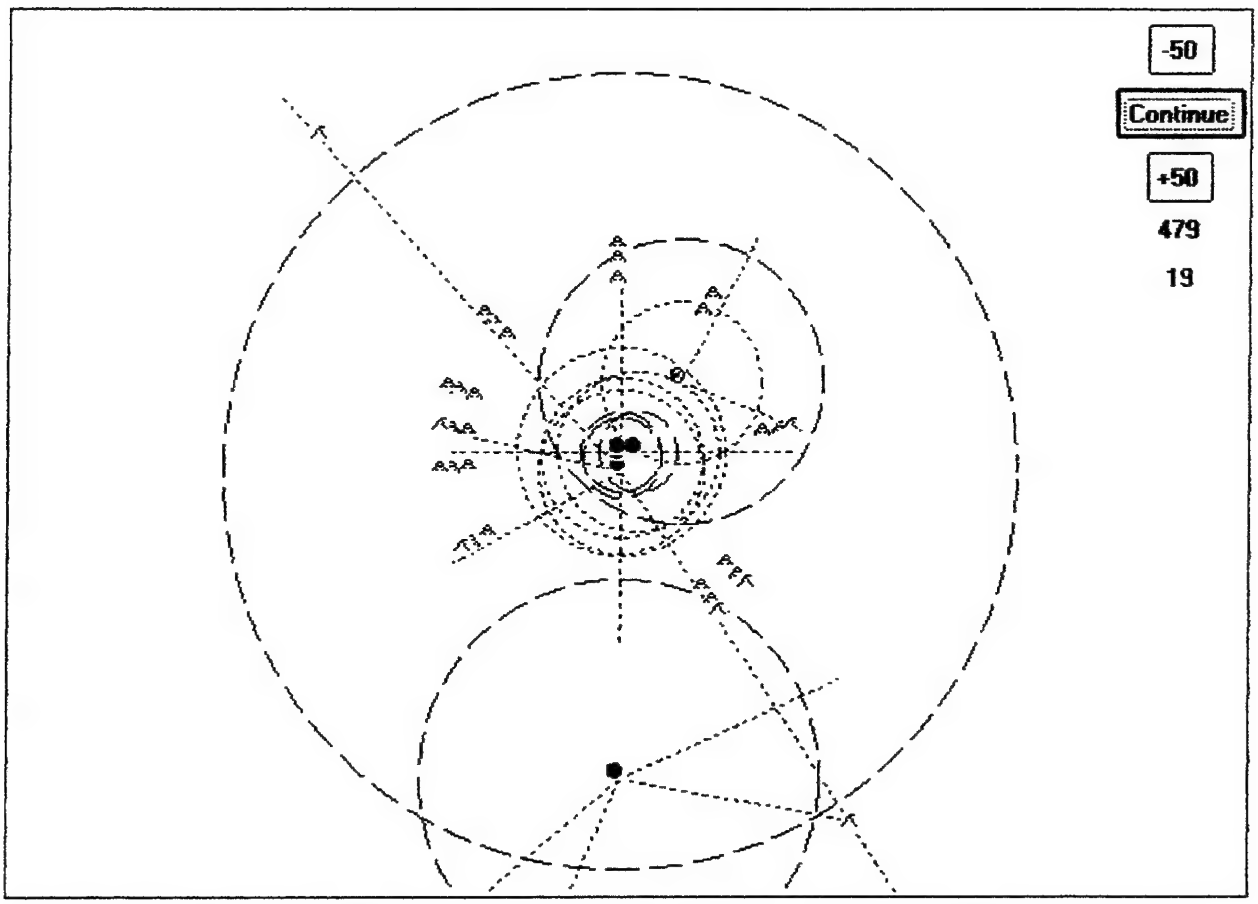

Figure (32). Battle group DTE simulation graphics example. Six AAW platforms are shown.

IV. PRACTICAL EXAMPLE

A. MOTIVATION

Battle Group Commanders are operating in a more demanding littoral environment, while having fewer ships to do the job. Our motivation in developing an example scenario was to create a battle group with realistic capabilities, and test it against a realistic threat force. Our prototype requires Rules of Engagement (ROE) such that any unidentified aircraft closing the battle group is considered hostile and engaged. We created our AAW platforms using unclassified data, and our threats use fictional parameters. Since the purpose our *BGSAMS* prototype was to validate a methodology concept, we believed our example was served in an unclassified forum.

B. CONFIGURATION TESTED

Our initial threat set consisted of a total of 6 strike fighters, each firing 2 ASCMs (Anti-Ship Cruise Missiles). A wave attack will saturate AAW systems more rapidly than a stream raid, so we started with a wave of 4 F-7 Supersonic & Low aircraft, each launching two Kidder missiles, followed by 2 F-7 aircraft, each launching 2 Poupon missiles. Threat parameters and list construction instructions are found in Table (1) and section B.2 in Chapter III.

Our battle group consists of a CV, an AEGIS cruiser, 2 *Spruance*-class destroyers, a *Perry*-class frigate, and a CAP section of 2 F-14's. The *Spruance* destroyers have different capabilities for detection and engagement, illustrating the sensitivity of our simulation to system parameters. The platform parameters are found in Table (2).

We developed our station lists by including stations that would allow a close, but comfortable formation. It seems logical to assume that protecting the CV from ASCMs requires a tight formation, so we tested stations ranging from 1 to 15 NM. These ranges were selected based on preliminary tests showing a diminished value of stationing ships beyond a 15 NM range, using the sensitivity analysis utility and a station list with ranges out to 50 NM. Since only 5 stations would be required for the final ship solution, excluding CAP, and the 5 stations

for each ship could easily be placed within a 15 NM radius, the station list was trimmed to reduce the single-platform DTE simulation run times.

We used the AAW platforms in the order given in Table (2), and the station lists found in Appendix A. The CV was tested only in *ZZ*; remaining ships were tested in the second list found in Appendix A; CAP were tested in the third station list. We created a formation using the stationing algorithm; the formation was tested using the same threat set used for data generation.

A list of the best stations for each platform and their associated expected leaker values is contained in Table (3). Stations are presented in order of rank, as produced by the stationing algorithm. The stationing algorithm actually operates on the probability of kill data created by the single-platform DTE simulation; inspection of the stationing algorithm's rank list will show the probability of kill values for each station. The stationing algorithm probability of kill data represents the probability that any given threat out of the tested set will be attrited, as discussed in Chapter II. The related expected leaker values are presented in Table (3) for consistency.

Our first sensitivity analysis tested the effects of removing one platform from the battle group. Two options were used; the first simply removed a platform without restationing the remaining ships; the second used a new formation. Since each platform has expected leaker data available, generating a new stationing algorithm formation requires only building a reduced battle group list, and selecting the platforms in that battle group for the stationing algorithm. User-designed formations can be tested without generating single-platform expected leaker data; single-platform data is only used by the stationing algorithm and plays no role in the battle group DTE simulation. Results of our example battle group simulation runs are presented in Table (4).

RANK	CV-41 Midway	DD-972 Oldendorf	CG-53 Mobile Bay	DD-991 Fife	FFG-38 Curts	CAP 2 x F-14
1	0 0*	1 170*	1 170	1 170	15 45*	50 180*
	8.198	6.667	7.042	8.53	10.427	9.574
2	Not Tested	3 180	3 180*	3 90*	12 165	50 135
	-----	7.227	7.361	9.414	10.467	9.628
3	-----	3 225	3 135	3 180	15 135	75 180
	-----	7.241	7.571	9.668	10.481	9.653
4	-----	3 45	3 225	3 135	15 150	50 225
	-----	7.574	7.582	9.695	10.481	9.707
5	-----	3 135	15 90	3 315	12 180	60 180
	-----	7.629	7.631	9.734	10.534	9.707

Table (3). Station rank list by platform. The upper entry gives the station, and the lower entry gives the associated expected leaker value. Initial stationing algorithm solution stations are marked with an asterisk. Stations are listed by range (NM) and bearing (deg) from *ZZ*.

RUN	CV-41 Midway	DD-972 Oldendorf	CG-53 Mobile Bay	DD-991 Fife	FFG-38 Curts	CAP 2 x F-14	Before Restationing	After Restationing
1	x	x	x	x	x	x	2.979	—
2	x		x	x	x	x	4.353	4.259
3	x	x		x	x	x	4.046	4.081
4	x	x	x		x	x	3.439	3.375
5	x	x	x	x		x	3.326	3.186
6	x	x	x	x	x		3.628	----

Table (4). Expected leaker values for modified battle groups.

In Table (4), platforms used for each run are marked with an 'x.' The original expected leaker values are the battle group DTE simulation results when platforms remained in their original stations after a loss. The last column of expected leaker values are those obtained by creating a new formation using the stationing algorithm, excluding the removed platform from consideration. Table (5) gives the stations used for each platform in the runs documented in Table (4). The stations listed for runs 2 through 5 are those which gave the values in the last column in Table (4). In each case, the difference after restationing was marginal, and not likely significant enough to conclusively show an improvement or degradation in AAW performance. Removing CAP did not allow an improved ship formation.

RUN	CV-41 Midway	DD-972 Oldendorf	CG-53 Mobile Bay	DD-991 Fife	FFG-38 Curts	CAP 2 x F-14
1	0 0	1 170	3 180	3 90	15 45	50 180
2	0 0	-------	1 170	3 180	15 45	50 180
3	0 0	1 170	-------	3 180	15 45	50 180
4	0 0	1 170	3 180	-------	15 45	50 180
5	0 0	1 170	3 180	3 90	------	50 180
6	0 0	1 170	3 180	3 90	15 45	------

Table (5). Stations used for simulation runs given by range, bearing from ZZ.

We also conducted a sensitivity analysis to test the saturation effect of increasing threat density. We used the original battle group in the stationing algorithm formation, increasing each attack by 2 supersonic F-7 fighters, each firing 2 supersonic Kidder missiles in the same manner as before. AAW platforms were not degraded or otherwise modified between attacks. The results of those attacks are presented in Table (6). Attack sizes are denoted by the number of supersonic aircraft plus the number of subsonic aircraft, each firing 2 missiles, as before.

Attack Size	4 + 2	6 + 2	8 + 2	10 + 2	12 + 2
E[Leakers]	2.979	5.401	7.639	10.476	13.532

Table (6). Battle group sensitivity to attacker density. Each attacker fires 2 ASCMs; *4+2* denotes 4 supersonic attackers followed by 2 subsonic attackers.

While the number of leakers values may seem high, they do not include self-defense weapons such as jamming, decoys, or CIWS (Close-In Weapons System) contributions. Nevertheless, the results have obvious trends, and we believe there is sufficient data to create concern over how we employ our AAW platforms. Certainly, our results depend heavily on our assumption that the detection and kill equations (2.4 and 2.5, respectively) are accurate enough. That may not be a valid assumption; the equations have not been accredited or compared to other accredited models. We believe that they are representative of general trends, however. Also important is that the vehicles we used do not necessarily have operating characteristics which accurately model real-world platforms. Threats were designed to produce noteworthy results; AAW platforms were modeled using conservative, unclassified parameters. Using more realistic data would undoubtedly influence our results; the differences in performance obtained by using different parameters for each of our *Spruance* destroyers shows the sensitivity of our model to parameter changes. Still, without more accurate data, no conclusions can be drawn about the fidelity of the detection and kill modeling used in our prototype.

With the above concerns in mind, some general considerations have emerged from our analysis. Certainly, the value added to AAW defense by AEGIS ships has been shown, even without including their full capabilities. We did not fully appreciate the role played by an AEGIS platform before seeing the simulation graphics as the threats were engaged *en masse* by our AEGIS cruiser. Without a doubt, AEGIS has a major role to play in littoral naval warfare.

AAW ships with modest area defensive capabilities, such as FFGs, add more to battle group defense when stationed away from the screen center. Dispersing those types of platforms allows their modest capabilities to have more of a net effect on the outcome, since they can contribute by attriting launch platforms before they can reach a firing position. Also, since they are limited to 1 missile-in-flight, modestly-capable ships add little to the close-in defense beyond adding another target for incoming missiles. While that reduces the probability that any particular platform will be targeted, it does not move toward the goal of preventing successful attacks.

Also, our methodology shows that highly-capable, short-range AAW platforms can contribute a significant amount to battle group defense against low-flying missiles. When attacking ASCMs approach at extremely low altitudes, the longer ranges and higher capabilities

of systems built for area AAW defense are diluted to a level which closely approximates a short-range AAW system. Considering platforms with short-range AAW missiles does have significant value. In fact, *Oldendorf* proved to be the most capable ship in a traditional *shotgun* role, behind the CV. This effect was definitely related to the flight profile of our threat set, and we cannot say that it translates into real-world capabilities since our model has not been accredited.

CAP have the most value at a modest range from *ZZ*. Using a single CAP station and modest capabilities, we were able to reduce the expected number of leakers by nearly 20 percent. As CAP are added to the battle group, similar magnitudes of leaker reduction can be expected, holding the threat constant. The CAP stationing problem can easily be accommodated by our model, which presents another important capability for AAW commanders.

We believe that our methodology deserves consideration for development into a deployed TDA. The next chapter presents our remaining conclusions, and our recommendations for deployment and enhancements of our method.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

Our AAW stationing methodology represents a step forward for AAW Tactical Decision Aids. It can provide commanders with a sound recommendation for countering a capable and potentially lethal threat. In the littorals, AAW stands out as a difficult problem no matter what potential adversary is faced.

A Navy-specific tool based on our methodology would be inexpensive, since the computing engine and the basic building blocks already exist in JMCIS (Joint Maritime Command Information System). Since the advent of modern SAM systems, tactical users have had to rely on laboratory recommendations for their employment, because insufficient analytical capability was available for effective tactic development at sea.

The Navy is rapidly moving toward building a capability to train and develop our operational art without leaving port. Simulation and modeling will undoubtedly play an even larger role than they already do in the life of an average sailor. We use simulations to train operators every day-- the ACTS (AEGIS Combat Training System) and OBT (On-Board Trainer) systems represent two significant training models that are used daily. But those models train operators to push buttons, not to maximize their combat system's potential. We need models which do more than teach system operation. This methodology represents one such example-- it can show users how to station a battle group to help defeat an enemy who is trying very hard to find and exploit weaknesses.

B. APPLICATIONS

This methodology has many applications. It could be used both in operational and force planning decision processes. Any commander facing the problem of distributing AAW assets could benefit from a tool based on the techniques developed here. The applications are not specific to naval AAW; land-based AAW has significant overlap in its methods and tactical philosophy.

1. JMCIS TDA

JMCIS has been adopted as the Navy's standard command and control computer platform. It began as an outgrowth of the Tomahawk weapons systems, as a means for non-Tomahawk-capable ships to provide input to the Tomahawk targeting system. It was soon discovered that JMCIS was a powerful tool for many other applications. It has become a workhorse platform of operations officers throughout the Fleet, offering a wide range of features which suit the development of this methodology directly. Detailed map displays, platform, emitter and weapons databases, and motion models all are available now. The computing engine would provide an order of magnitude increase in speed over the desktop PC platform used by the prototype model. Most importantly, the JMCIS computers and software are constantly improved, so the addition of a new TDA would not require an entirely new contract or development of platform requirements.

2. AWS TDA

Another possible platform for an AAW stationing TDA is the AEGIS Weapons System (AWS). The use of more powerful computers in the newest versions would allow the inclusion of a capability of this sort as an integral part of the Navy's premiere combat system.

3. TBMD

This methodology could be extended to include Theater/Tactical Ballistic Missile Defense forces as well. While the methods of engaging TBMs are different from traditional AAW, they are easily modeled. A tool could be developed for testing various locations around a protected area, offering an ability to develop a force disposition in much the same manner as developing a formation using the stationing algorithm.

4. PATRIOT / Overland AAW

Land-based AAW forces would also benefit from the use of a stationing aid. While specific tactical guidelines for the placement of air defenses around a protected area do exist, it

is suspected that operational situations do not always lend themselves to using the guidelines provided. A land-based stationing tool could help account for specific situations in much the same manner as a naval AAW tool would. The inclusion of map data for RADAR coverage models could easily be added, giving an ability to model terrain masking and other overland considerations.

5. Joint AAW Coverage Model

A joint naval/overland AAW coverage model could be developed for theater-level use in combined operations. Such a tool could help combine available AAW forces to maximize their defensive potential. Solving this operational problem will be important as soon as the next generation of 200+ NM SAMs is deployed. Coordination of sensors and weapons systems will become increasingly complex as each contributing service has longer-range weapons and sensors available to counter the enemy. Including detailed sea- and land-based environmental models, as well as terrain data, would be critical to the usefulness of this type of model, since forces would not be operating in a homogeneous environment. With those considerations, a Joint model could be developed, starting with the methods presented here.

6. Classroom Training Aid

The methodology could be developed into a classroom training aid, useful in refining the concepts of battle group AAW. Although it would not focus on an operator-level perspective, it would allow the consideration of the parameters important to solving the problem of ship and CAP stationing, and provide a tool useful for the investigation of system casualty effects on battle group AAW performance. A tool like this would train users to think in terms of the larger tactical picture, instead of focusing solely on how to solve the AAW problem by considering only one unit out of many.

C. STATIONING ALGORITHM IMPROVEMENTS

A number of improvements should be included in a deployable stationing algorithm. They would focus on providing a more constrained model which offers users an ability to control

the solution generated more closely. While much of the added value of these constraints could probably be gained by manually altering the stationing algorithm's solution, there may be benefits to adding them which are not immediately obvious. The following constraints should be considered.

1. Quadrant Covering Constraints

A quadrant covering constraint would include a requirement for coverage in specified quadrants. It would prevent solutions which are biased toward the main threat axis. One limitation of the prototype is that its solutions for less-capable platforms tend to place them along the least-capable threat axis, where they can make easier kills. This provides more depth of fire along that axis, but leaves others exposed. While this solution makes sense, given the solution method, it might not make sense to leave certain areas vulnerable in the real world.

2. Spacing Constraints

A spacing constraint would allow the specification of a minimum or maximum station distance spacing in the final solution. Understandably, commanders at sea are not often comfortable operating in close quarters for long periods of time.

D. SIMULATION FIDELITY IMPROVEMENTS

The simulation models could benefit from a number of improvements. Since those used in the prototype have not been validated or accredited, those steps must be investigated. It is believed by the author that the models developed here would compare favorably to those already in operational use. After studying the models used in Operation Desert Storm, details based on lessons learned for AAW modeling were added to the prototype simulations. [Ref. 8] Still, other improvements could be made. This section discusses a number of details worth investigating. The improvements are given in an order of preference, based on their additional requirements. Items located far down the list would add little or no value to a model without including those of higher priority. Added details mean little if they do not account for all of the main variables which influence their use. Returning to an earlier example, increased flight path

modeling fidelity will mean litle if it has no influence on the actions taken by the engaging platforms. Therefore, the inclusion of additional details should be carefully studied before committing to them.

1. Geographical Database Links

One of the first additions should be the inclusion of geographical data in the simulations. This would allow threat profile and station list construction based on actual locations, greatly increasing the practicality of the models.

2. Meteorological Model Improvements

While the prototype was designed around a detection model that uses 95% probability of detection ranges, it does not include provisions for surface ducting effects or air mass changes, as often encountered at the land-sea interface. Including a more accurate meteorological model would enhance model fidelity, especially for littoral operations. Since the goal of this project was to develop a methodology useful in littoral warfare, this feature is considered important. Including it in tandem with geographical modeling would be a natural combination, and a worthwhile addition. A natural inclusion would be the IREPS model in the detection module of the simulations. That model is present in current JMCIS software versions.

3. AAW Platform Motion

Another addition could be a capability for AAW platform motion during the simulation process. Due to the relative speeds, the most likely benefactors of that extension would be CAP or other aircraft. But ships might benefit if short-distance maneuvers are considered, such as those to allow a better shot at a threat that is masked by another platform. This assumes that another important feature would be added, which would prevent ships from firing over (or through) neighboring platforms. The prototype does not include that provision.

4. Multiple Threat Target Selection

An ability to model contemporary seeker-head designs would allow more than one possible target location in a simulation. Targets could be selected randomly from within the threat's seeker envelope; that provision would also enhance soft-kill modeling, such as jamming and decoys.

Including this feature would require a more detailed motion model for threats, since they would change course during their flight, possibly a number of times. Adding to the motion model detail would allow more detailed route planning for threats, but it would also increase the time requirements of developing a scenario. A TDA designed for fast results in an operational environment might not prove useful with such a high level of detail.

5. Multiple AAW System Modeling per Platform

Developing a method of selecting multiple potential targets would give benefits to including multiple weapons system models on each platform. At this level of detail, the model faces a danger of being adopted as a prediction aid, rather than a stationing tool. Still, these improvements could be beneficial if they are kept in context.

6. Threat Flight Profile Modeling

The inclusion of short-range active AAW systems would increase the need for threat flight profile fidelity, since shorter ranges will be more influenced by small changes than would longer-range systems. For practical purposes, short-range systems could be considered as those with a maximum effective range of less than 6 NM, since that is the practical minimum range for area defense weapons.

DTE sequences for ranges less than 4 NM will typically last 5 seconds or less; threats at a 9 NM range facing the same reaction times will allow 15 or more seconds for engagements; threats at 20 NM allow at least 30 seconds. With typical sweep intervals of less than 1 second, a 15 second time period using the engagement criteria already present in the prototype model is sufficient for engagement (at 9 NM) and destruction of threats at up to 6 NM from the engaging

platform. Closer threats require a different set of engagement criteria, leading to the definition of short-range systems presented above.

Quick engagements will be more affected by radical target maneuvers, explaining the need for increased flight path fidelity in a short-range AAW model. The detection and kill functions would need to account for acceleration and more accurate aspect angles, at a minimum.

7. ECM/Decoy Modeling

Work by Schulte demonstrates the value of soft-kill capabilities in AAW [Ref. 9]. It has been shown to be nearly 100 percent effective when properly employed, adding a significant defensive capability. Soft kill modeling would add to the model, if the measures above were also included. It would not be worthwhile to add this capability without increasing the hard-kill modeling capabilities, since soft-kill measures are typically employed in tandem with hard-kill weapons.

Soft kill modeling would require AAW platform motion and wind modeling, since decoys such as chaff are strongly affected by both factors. At this level of detail, it is guessed that the simulation speed of a TAC-3 JMCIS workstation would be approximately equal to that of the prototype model presented here (using a *Pentium*-based PC), although at a significantly higher fidelity level.

This level of fidelity would be worth considering, since there are few useful tools available to Fleet users which teach the effects and value of countermeasures use. The model could reveal when using decoys like chaff is a bad idea, or when it is absolutely necessary.

8. IFF Modeling

One feature notably absent in the prototype is IFF (Identification Friend-or-Foe) modeling. Since this model does not include friendly air vehicles beyond CAP, it would not benefit from IFF modeling. If other friendly vehicles were added, perhaps commercial air traffic or military aircraft carrying out other missions, then IFF modeling would add value to the model.

Adding IFF would again increase the modeling complexity. It would require a set of decision rules for target identification, a density function for IFF reliability (unfortunately, it is not 100% reliable), and a set of decision rules for weapons status equating to rules of engagement. The prototype model avoids these complications by focusing only on capabilities and assuming a warning red, weapons free environment, and by keeping the friendly and threat vehicles in separate lists, preventing blue-on-blue engagements.

9. Other Modifications

This list certainly is not exhaustive. It represents the fidelity enhancements most obvious to the author, and includes only those related to operational problems and capabilities with which he is familiar. Many more operational problems could be examined for possible modeling value. Also, doctrinal and engineering problems could be studied, including sensor channelization effects, combat systems track file database limitations, and even the effects of conducting the AAW mission while pursuing other missions could be modeled. The problems worth studying are limited only by the experience and imagination of the users and modelers.

E. INTERFACE IMPROVEMENTS

A number of improvements could be made in the user interface for a deployable system. Any improvements should focus on making the software easier to use. A number of examples have been provided here.

1. Threat Flight Path Creation

Threat flight path creation could be simplified by allowing a point-and-click designation of the flight path intermediate points. This feature would be very useful if geographical data was added to the simulation. With that, users could create flight paths that correspond to real-world scenarios; threat sectors could be related to terrain features.

2. Station List Construction

Station list creation also would benefit from a point-and-click designation method. Users could relate stations graphically instead of being required to make a translation from a visual picture to a numerical representation of it, as is required in the prototype model. This improvement would speed the creation of station lists considerably.

3. Simulation Graphics

Simulation graphics presentation would benefit from making more data available during a simulation run. For example, an interface more similar to a contemporary NTDS console would be a good model to follow. Users of that system have the ability to select individual vehicles and inspect current data related to that vehicle. While these types of features would not add value to the model's stationing algorithm, they would add value to the tool if it were used as a training aid.

LIST OF REFERENCES

1. Naval Doctrine Publication 1, *Naval Warfare*. Department of the Navy, March 1994.

2. Joint Publication 1, *Joint Warfare of the U.S. Armed Forces*. Joint Chiefs of Staff of the United States, November 1991.

3. Naval Warfare Publication 32, *Anti-Air Warfare*, Revision L, July 1992.

4. Naval Warfare Publication 10-1-21, *Anti-Air Warfare Commander's Manual*, Revision A, August 1990.

5. Law, Averill M. and Kelton, W. David. *Simulation Modeling and Analysis.* Second Edition, p. 358, McGraw-Hill, Inc. 1991.

6. Lawler, Eugene L. *Combinatorial Optimization: Networks and Matroids*. Holt, Rinehart and Winston, 1976.

7. Ahuja, Ravindra K., Magnanti, Thomas L., and Orlin, James B. *Network Flows. Theory, Algorithms, and Applications.* Prentice-Hall, 1993.

8. Case, F.T., "Analysis of Air Operations During Desert Shield/Desert Storm," United States Air Force Studies & Analyses Agency, Regional Forces Division, November 1991.

9. Schulte, John C. "An Analysis of the Historical Effectiveness of Anti-Ship Cruise Missiles in Littoral Warfare," Naval Postgraduate School Master's Thesis, Monterey, CA, September 1994.

BIBLIOGRAPHY

FON AZ3053-1-90, *DD-963 Class Destroyer SDSMS (Baselines 6 and 3- NTDS) AAW Tactical Doctrine,* COMSURFWARDEVGRU, July 1990.

FON AZ3053-2-90, *Tactical Employment of the New Threat Upgrade (NTU) Medium Range AAW Combat System,* Revision A, COMSURFWARDEVGRU, December 1990.

LL AZ3011-1-80, *Battle Group Anti-Air Warfare*, FITRON 41, September 1980.

Naval Warfare Publication 63, *Surface Ship Anti-Air Warfare Tactics*, January 1992.

Naval Warfare Publication 12-5-1, *Tactical Action Officer Handbook*, April 1993.

Naval Warfare Publication 12-2, *Tactical Threat to Surface Naval Forces*, May 1993.

TAM AZ3011-1-88, *AAW Stationing of AEGIS and NTU Ships*, COMTHIRDFLT, March 1988.

TAM AZ3020-1-90, *AOE/AOR Class Ship SDSMS AAW Operations*, COMSURFWARDEVGRU, October 1990.

TAM AZ3030-1-92, *Planning Guide for Medium-Range AAW (MRAAW)*, COMTHIRDFLT, June 1992.

TAM AZ3051-1-91, *Carrier Self-Defense Surface Missile System AAW Operations,* COMSURFWARDEVGRU, September 1993.

TAM AZ3052-2-92, *Tactical Employment of the New Threat Upgrade (NTU) Extended Range AAW Combat System,* Revision B, COMSURFWARDEVGRU, October 1991.

TAM AZ3052-2-90, *Employment of AN/SPY-1A/B RADARs Against Low Observable Targets,* COMSURFWARDEVGRU, September 1990.

TAM AZ3054-1-92, *Tactical Employment of the FFG-7 MK 92 Mod 2 AAW Weapon System,* Revision A, COMSURFWARDEVGRU, November 1991.

TAM AZ3054-2-90, *Tactical Employment of the FFG-7 MK 92 Mod 6 AAW Weapon System,* COMSURFWARDEVGRU, October 1990.

APPENDIX. STATION FILES

Below are the station lists used in the thesis example problem. The first list is stored as "C:\BGSAMS\STATIONS\ZZ." The second list of stations is stored as "C:\BGSAMS\STATIONS\BGTEST.1", and the third list of stations is stored as "C:\BGSAMS\STATIONS\CAPTEST.1" Each station is contained on a single line. The first number is the distance in NM from ZZ. The second number is the bearing in degrees from ZZ.

File: ZZ

<Beginning of file>

0 0

<End of File>

File: BGTEST.1

<Beginning of file>

1 170
3 270
3 315
3 0
3 45
3 90
3 180
3 135
3 225
5 0
5 180
5 270
5 90
5 135
5 225
5 315
5 45
10 0
10 180
10 270
10 90
10 45
10 135
10 225

10 315
12 165
12 180
12 195
15 165
15 0
15 30
15 45
15 60
15 90
15 120
15 135
15 150
15 180
15 210
15 225
15 270
15 300
15 315
15 330

<End of File>

File: CAPTEST.1

<Beginning of File>

40 135
40 180
40 225
50 0
50 135
50 180
50 225
60 0
60 135
60 180
60 225
75 0
75 135
75 180
75 225
100 0
100 135
100 180
100 225
<End of File>

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center..........2
   Cameron Station
   Alexandria, Virginia 22304-6145

2. Library, Code 52..........2
   Naval Postgraduate School
   Monterey, California 93943-5101

3. Professor David Schrady, Code OR/SC..........2
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California 93943-5000

4. Professor Michal Bailey, Code OR/BA..........2
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California 93943-5000

5. Professor Wayne Hughes, Code OR/HI..........2
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California 93943-5000

6. Professor George Conner, Code OR/CU..........1
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California 93943-5000

7. Professor Paul Bloch, Code OR/BC..........1
   Department of Operations Research
   Naval Postgraduate School
   Monterey, California 93943-5000

8. Professor Charles Calvano, Code ME/CA..........1
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, California 93943-5000

9. Mr. Robert Wood..........1
   Director, Center for Naval Warfare Studies
   U.S. Naval War College
   Newport, Rhode Island 02841

10. Dr. Roger Barnett..1
Joint Operations Division, Code 12
U.S. Naval War College
Newport, Rhode Island 02841

11. Commanding Officer...1
Tactical Training Group, Atlantic
Fleet Combat Training Center, Atlantic
Dam Neck
Virginia Beach, Virginia 23461

12. Commanding Officer...1
Tactical Training Group, Pacific
ATTN: CAPT Neil Byrne, USN
Fleet Combat Training Center, Pacific
San Diego, California 92147

13. Commanding Officer...1
Tactical Training Group, Atlantic
Fleet Combat Training Center, Atlantic
Dam Neck
Virginia Beach, Virginia 23461

14. Mr. Ib Hanssen..1
NSWC Carderock Division, Code 174
Bethesda, Maryland 20084-5000

15. Mr. John V. Hall..1
Center for Naval Analyses
4401 Ford Avenue, Box 16268
Alexandria, Virginia 22302-0268

16. VADM William Rowden, USN (Ret)...1
Center for Naval Analyses
4401 Ford Avenue, Box 16268
Alexandria, Virginia 22302-0268

17. Mr. W.J. Hurley...1
Institute for Defense Analysis
1801 North Beauregard Street
Alexandria, Virginia 22311-1772

18. Dr. Frank Shoup..1
    Expeditionary Warfare Division
    Office of the CNO
    Pentagon Room 4A720
    Washington, D.C. 20350-2000

19. Dr. Gary Grittner...1
    Surface Warfare Division
    Office of the CNO
    Pentagon Room 4C510
    Washington, D.C. 20350-2000

20. CDR Aland..1
    Command and Control Division (N62)
    Office of the CNO
    Pentagon
    Washington, D.C. 20350-2000

21. CAPT Edward C. McDonough...1
    Staff, CINCPACFLT (N-80)
    Pearl Harbor, Hawaii 96860

22. Commanding Officer..1
    USS Mobile Bay (CG-53)
    FPO AP 96672-1173

23. Commanding Officer..1
    USS Bunker Hill (CG-52)
    FPO AP 96693

24. LT D.D. Gabrielson, USN...2
    1618 Mishawaka Road
    Grand Rapids, Minnesota 55742